

**PENGEMBANGAN SISTEM INFORMASI *SALES FORCE*  
*AUTOMATION* DAN *SERVICE AUTOMATION* PEMESANAN  
DAN PENJUALAN MENGGUNAKAN METODE *RATIONAL*  
*UNIFIED PROCESS*  
(STUDI PADA KANTIN FILKOM UNIVERSITAS BRAWIJAYA)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Helmy Danu Prameireza  
NIM: 145150400111094



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI SALES FORCE AUTOMATION DAN SERVICE  
AUTOMATION PEMESANAN DAN PENJUALAN MENGGUNAKAN METODE  
RATIONAL UNIFIED PROCESS  
(STUDI PADA KANTIN FILKOM UNIVERSITAS BRAWIJAYA)

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Helmy Danu Prameireza  
NIM: 145150400111094

Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Juli 2018

Telah diperiksa dan disetujui oleh:

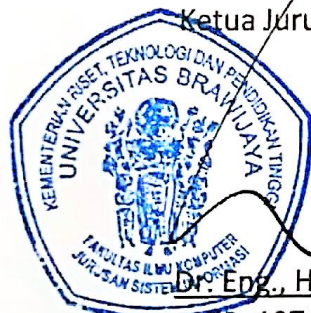
Dosen Pembimbing I

Dosen Pembimbing II

Yusi Tyroni Mursityo, S.Kom., M.AB.  
NIP: 19800228 200604 1 001

Djoko Pramono, S.T., M.Kom.  
NIP: 19780108 200501 1 002

Mengetahui  
Ketua Jurusan Sistem Informasi



Dr. Eng., Herman Tolle, S.T., M.T  
NIP: 19740823 200012 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah proposal skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “Pengembangan Sistem Informasi *Sales Force Automation* dan *Service Automation* Pemesanan dan Penjualan Menggunakan Metode *Rational Unified Process* (Studi Pada Kantin Filkom Universitas Brawijaya)”.

Dalam proses pengerjaan penelitian ini, penulis banyak mendapatkan ilmu tidak pernah didapatkan pada materi perkuliahan. Selain itu dalam penyusunan dan penulisan laporan ini penulis juga mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB, selaku dosen Pembimbing 1 yang senantiasa membimbing, memberikan motivasi dan saran kepada penulis selama penyusunan laporan skripsi.
2. Bapak Djoko Pramono, S.T., M.Kom., selaku dosen pembimbing 2 yang telah mengajarkan ketelitian serta banyak memberikan ilmu, saran, dan motivasi untuk penyelesaian skripsi ini.
3. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D. selaku dosen penasehat akademik yang senantiasa memberikan masukan dan solusi permasalahan akademik.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya Malang.
6. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
7. Ibu, Ayah serta keluarga penulis yang senantiasa memberikan semangat, motivasi dan dukungan untuk menyelesaikan semua tugas dan tanggung jawab penulis.
8. Ibu Tuti Edy Santoso selaku pengelola kantin filkom yang telah mengizinkan dan membantu penulis dalam menyelesaikan skripsi ini.
9. Mas Yhouga Evan, Mas Ageng, Mas Falachudin, Mas Nafi, serta kakak tingkat penulis yang telah memberi motivasi dan dukungan kepada penulis.
10. Jawara, Edwin, Adit Ramadhan, Tedja, Fikri, Fatih, Rosalia (Ocha), Iqbal, Windha, Raga, Titis, Sekar, Nanda Adhi, Savira, MON, Hilal, Aisyatul, Prana, serta sahabat-sahabat penulis lainnya yang telah banyak memberikan bantuan dan semangat.
11. Sarah, Ain, Arda, Ghany, Asdar, Andri Ginting, serta adik-adik tingkat penulis yang memberikan semangat kepada penulis.

12. Teman-teman UKM Nol Derajat Film UB yang telah rela membantu penulis ketika dalam kesulitan.

Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan baik format laporan maupun isinya. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan selanjutnya. Semoga laporan skripsi ini dapat bermanfaat bagi pembaca dan penulis.

Malang, 27 Juli 2018

Penulis

danupramei@student.ub.ac.id



## ABSTRAK

Saat ini di banyak instansi telah menerapkan sistem informasi bahkan usaha-usaha kecil seperti kantin telah mulai menerapkan sistem informasi didalamnya. Kantin Fakultas Ilmu Komputer (Fikom) sendiri saat ini telah memiliki teknologi untuk menerapkan sistem informasi, namun hanya sistem informasi pada kasir kantin. Pada sistem informasi yang telah diterapkan saat ini terdapat beberapa permasalahan yang dapat mempengaruhi kepuasan pembeli yaitu, alur proses bisnis pemesanan memakan waktu yang cukup lama dan penjual yang kebingungan dalam mengantarkan pesanan. Maka dari itu diperlukannya suatu pengembangan sistem yang dapat membantu menangani masalah tersebut.

Penelitian ini menggunakan modul pada operasional *Customer Relationship Management (CRM)*, *Sales force automation (SFA)* dan *Service Automation (SA)*. Metode pengembangan yang digunakan adalah *Rational Unified Process (RUP)* yang merupakan kerangka kerja dalam pengembangan sistem informasi yang menggunakan teknik iteratif dan inkremental. Metode RUP terdapat beberapa fase dalam melakukan pengembangan sistem, fase pertama yang dilakukan adalah fase inepsi, lalu dilanjutkan dengan fase elaborasi, fase konstruksi dan diakhiri dengan fase transisi. Implementasi sistem ini berbasis perangkat bergerak dengan platform android. Hasil pengujian *black-box* dengan 47 *test-case* dinyatakan berhasil, pengujian *compatibility* dengan 12 perangkat yang berbeda menggunakan *Test Lab* dan dari pengujian tersebut dinyatakan berhasil dijalankan semua serta pada *user accepted testing* yang melibatkan 30 responden pembeli dengan total 240 jawaban dan 1 responden pihak kantin dengan total 7 jawaban, hasilnya menunjukkan bahwa tingkat penerimaan dari pembeli adalah 85,92% dan 82,86% pada sisi kantin Fikom. Pada Pengujian Waktu dengan perbandingan proses bisnis *as-is* dengan proses bisnis *to-be*, dapat disimpulkan bahwa pada proses bisnis *to-be* lebih cepat 4 menit 30 detik dari proses bisnis *as-is*. Dari hasil pengujian tersebut dengan menerapkan sistem informasi ini dapat mempermudah dan mempercepat alur pemesanan dan penjualan pada Kantin Fikom.

Kata kunci: *Customer Relationship Management (CRM)*, *Sales force automation (SFA)*, *Service Automation (SA)*, *Rational Unified Process (RUP)*, *Android*, *sistem informasi*, *kantin*.



## ABSTRACT

*Currently in many agencies have implemented information systems even small businesses such as canteen have begun to implement information systems in it. Canteen Faculty of Computer Science (Filkom) currently has the technology to implement information systems, but only the information system at the canteen cashier. There are some problems that had been found in the current information system canteen FILKOM. These problems can affect customer's satisfaction, for instance the business process of food order takes a long time and the sellers are confused while delivering the order. Hence there is a need for developing a system which can overcome these problems.*

*This study uses modules on the operational Customer Relationship Management (CRM) Sales force automation (SFA) and Service Automation (SA). Rational Unified Process (RUP) is a framework which will be used for developing the information system which has iterative and incremental techniques. RUP method has several phases, the first phase is inception phase, then elaboration phase, construction phase and ending with transition phase. The implementation of the system is mobile device-based with android platform and firebase as the database server. The results of black-box testing with 47 test-cases were succeeded, compatibility testing with 12 different devices using Test Lab was successfully executed and User Acceptance Testing (UAT) shows that the value of acceptance from the buyer is 85,92% and 82,86% on the side of the FILKOM canteen. The conclusion of the test results that is by applying this information system at canteen FILKOM can facilitate ordering and sales.*

**Keywords:** *Customer Relationship Management (CRM), Sales force automation (SFA), Service Automation (SA), Rational Unified Process (RUP), Android, information system, canteen.*

## DAFTAR ISI

KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN .....	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	4
1.4 Manfaat.....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Kantin .....	9
2.2.1 Kantin Fakultas Ilmu Komputer Universitas Brawijaya .....	9
2.3 <i>Customer Relationship Management</i> .....	10
2.3.1 Operasional CRM.....	10
2.4 Sistem Informasi .....	13
2.5 Proses Bisnis.....	14
2.5.1 <i>Business Process Model and Notation (BPMN)</i> .....	14
2.6 Analisis Kebutuhan Sistem Informasi .....	20
2.7 Perancangan Sistem Informasi .....	22
2.8 <i>Unified Modelling Language (UML)</i> .....	23
2.9 Implementasi Sistem Informasi .....	27
2.10 Android .....	28
2.10.1 Java.....	29
2.10.2 Android Studio .....	29



2.10.3 Fundamentals .....	30
2.11 Firebase.....	33
2.11.1 Firebase Real Time Database .....	33
2.12 Pengujian Sistem Informasi .....	33
2.12.1 Black-Box Testing .....	34
2.12.2 Compatibility testing .....	34
2.12.3 User Acceptance Testing .....	35
2.13 Skala Likert .....	37
2.14 Rational Unified Process .....	38
BAB 3 METODOLOGI .....	40
3.1 Studi Literatur .....	41
3.2 Pengumpulan Data .....	41
3.2.1 Wawancara .....	41
3.2.2 Observasi.....	41
3.3 Fase <i>Inception</i> .....	41
3.3.1 Pemodelan Proses Bisnis.....	42
3.3.2 Perbaikan Proses Bisnis.....	42
3.4 Fase <i>Elaboration</i> .....	42
3.4.1 Analisis Kebutuhan.....	42
3.4.2 Perancangan.....	43
3.5 Fase <i>Construction</i> .....	43
3.5.1 Implementasi .....	43
3.5.2 Pengujian Sistem .....	43
3.6 Fase <i>Transition</i> .....	44
3.6.1 User Acceptance Testing .....	44
3.6.2 Pengujian Waktu .....	44
3.7 Kesimpulan.....	44
BAB 4 ANALISIS KEBUTUHAN .....	45
4.1 Business Modelling .....	45
4.1.1 Pemodelan Proses Bisnis As-Is .....	45
4.1.2 Pemodelan Proses Bisnis To-be .....	48
4.1.3 Analisa Kebutuhan Pemangku Kepentingan .....	52

4.2 Requirement .....	53
4.2.1 Identifikasi Aktor .....	53
4.2.2 <i>User Requirement</i> .....	54
4.2.3 Fitur Sistem Informasi .....	55
4.2.4 <i>Software Requirement Specification</i> .....	55
4.2.5 Use Case Diagram .....	57
4.2.6 <i>Use Case Scenario</i> .....	59
BAB 5 PERANCANGAN SISTEM.....	70
5.1 <i>Activity Diagram</i> .....	70
5.2 <i>Sequence Diagram</i> .....	75
5.3 <i>Class Diagram</i> .....	80
5.4 <i>Physical Data Model</i> .....	82
5.5 Perancangan Antarmuka Pengguna (UI).....	83
BAB 6 IMPLEMENTASI SISTEM .....	88
6.1 Iterasi Ke-1 Fase <i>Elaboration</i> .....	88
6.1.1 Spesifikasi Sistem .....	88
6.1.2 Batasan Implementasi.....	89
6.2 Fase <i>Construction</i> .....	90
6.2.1 Implementasi Basis Data.....	90
6.2.2 Implementasi Kelas .....	92
6.2.3 Implementasi Kode Program .....	94
6.2.4 Implementasi Antarmuka Pengguna (UI).....	110
BAB 7 PENGUJIAN SISTEM .....	120
7.1 Pengujian <i>Black Box</i> .....	120
7.1.1 Rancangan Pengujian <i>Black Box</i> .....	120
7.1.2 Hasil Pengujian <i>Black Box</i> .....	128
7.2 Pengujian <i>Compatibility</i> .....	133
7.3 <i>User Acceptance Testing</i> .....	133
7.3.1 Hasil <i>User Acceptance Testing</i> pada Pembeli .....	136
7.3.2 Hasil <i>User Acceptance Testing</i> pada Pihak Kantin.....	138
7.4 Pengujian Waktu .....	140
BAB 8 PENUTUP .....	141

8.1 Kesimpulan.....	141
8.2 Saran .....	142
DAFTAR PUSTAKA.....	143
LAMPIRAN A HASIL WAWANCARA.....	145
LAMPIRAN B HASIL UAT .....	147



## DAFTAR TABEL

Tabel 2.1 Rangkuman Kajian Pustaka .....	7
Tabel 2.2 Notasi BPMN .....	15
Tabel 2.3 Simbol-simbol Diagram Use Case .....	23
Tabel 2.4 Keterangan Simbol-Simbol Activity Diagram .....	25
Tabel 2.5 Simbol-Simbol Diagram Sequence .....	26
Tabel 2.6 Simbol-Simbol Class Diagram .....	27
Tabel 2.7 Rincian Bobot Nilai .....	37
Tabel 4.1 identifikasi permasalahan dan waktu .....	47
Tabel 4.2 Analisa Domain Masalah dan Solusinya .....	47
Tabel 4.3 Tipe Pemangku Kepentingan.....	52
Tabel 4.4 Peran dan Perwakilan Pemangku Kepentingan .....	52
Tabel 4.5 Peran Pengguna.....	53
Tabel 4.6 indentifikasi Aktor .....	53
Tabel 4.7 Kebutuhan Pengguna .....	54
Tabel 4.8 Fitur Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom .	55
Tabel 4.9 SRS Functional .....	55
Tabel 4.10 SRS Non-Functional .....	57
Tabel 4.11 Use case scenario Melihat Pesanan .....	59
Tabel 4.12 Use case scenario Edit Status pesanan selesai.....	60
Tabel 4.13 Use case scenario Melihat Menu Penjualan .....	60
Tabel 4.14 Use case scenario Memesan Produk Penjualan.....	61
Tabel 4.15 Use case scenario Membayar Pesanan .....	62
Tabel 4.16 Use case scenario Melihat Profil Akun (tambahan pada iterasi pertama fase elaborasi) .....	63
Tabel 4.17 Use case scenario Mengisi Saldo E-Dompet .....	63
Tabel 4.18 Use case scenario Melihat Hasil Penjualan (tambahan pada iterasi ke-1 fase elaborasi) .....	64
Tabel 4.19 Use case scenario Mengelola Produk Penjualan.....	65
Tabel 4.20 Use case scenario Tambah Produk.....	65
Tabel 4.21 Use case scenario Hapus Produk.....	66

Tabel 4.22 Use case scenario Update Produk.....	67
Tabel 4.23 Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi).....	67
Tabel 4.24 Use case scenario Edit status pesanan dibatalkan (tambahan pada iterasi ke-1 fase konstruksi) .....	68
Tabel 4.25 Use case scenario Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi) .....	68
Tabel 6.1 Spesifikasi Perangkat Keras Komputer .....	88
Tabel 6.2 Spesifikasi Perangkat Keras Perangkat Bergerak .....	88
Tabel 6.3 Spesifikasi Perangkat Keras Server .....	89
Tabel 6.4 Spesifikasi Perangkat Lunak Pengembangan .....	89
Tabel 6.5 Berkas Kelas <i>Model</i> .....	92
Tabel 6.6 Berkas Kelas <i>View</i> .....	92
Tabel 6.7 Berkas Kelas <i>Controller</i> .....	93
Tabel 6.8 Berkas Kelas <i>Physical Data Model</i> .....	94
Tabel 6.9 Kode Program Fungsi savePesanan() .....	94
Tabel 6.10 Kode Program Fungsi generatedKodePesanan().....	96
Tabel 6.11 Kode Program Fungsi insertPesanan() .....	97
Tabel 6.12 Kode Program Fungsi updatePesananBelumBayar() .....	97
Tabel 6.13 Kode Program Fungsi insertProdukPesanan().....	98
Tabel 6.14 Kode Program Fungsi updateStokProduk() .....	98
Tabel 6.15 Kode Program Fungsi getPesananBelumBayar().....	99
Tabel 6.16 Kode Program Fungsi getSaldo() .....	100
Tabel 6.17 Kode Program Fungsi bayar() .....	100
Tabel 6.18 Kode Program Fungsi potongSaldo().....	100
Tabel 6.19 Kode Program Fungsi updateStatusPesanan() .....	101
Tabel 6.20 Kode Program Fungsi generatedDepositPlus() .....	101
Tabel 6.21 Kode Program Fungsi inserTransaksiDeposit().....	102
Tabel 6.22 Kode Program Fungsi getTransaksi().....	102
Tabel 6.23 Kode Program Fungsi notifBank() .....	103
Tabel 6.24 Kode Program Fungsi ubahStatus().....	104
Tabel 6.25 Kode Program Fungsi tambahSaldo().....	104
Tabel 6.26 Kode Program Fungsi setStatusPesanan() .....	105

Tabel 6.27 Kode Program Fungsi tampilDialogFormUP() .....	106
Tabel 6.28 Kode Program Fungsi updateProduk() .....	107
Tabel 6.29 Kode Program Fungsi insertProduk() .....	108
Tabel 6.30 Kode Program Fungsi hapusProduk() .....	108
Tabel 6.31 Kode Program Fungsi editToko() .....	109
Tabel 6.32 Kode Program Fungsi tampilDialogPengambilan() .....	109
Tabel 7.1 Perancangan Pengujian <i>Black-Box</i> .....	120
Tabel 7.2 Hasil Pengujian <i>Black Box</i> .....	128
Tabel 7.3 Skenario task <i>User Acceptance Testing</i> pada Pembeli .....	134
Tabel 7.4 Skenario task <i>User Acceptance Testing</i> pada Pihak Kantin .....	134
Tabel 7.5 Pernyataan UAT untuk Pembeli .....	135
Tabel 7.6 Pernyataan UAT untuk Pihak Kantin .....	136
Tabel 7.7 Hasil <i>User Acceptance Testing</i> pada Pembeli.....	137
Tabel 7.8 Penilaian Bobot <i>User Acceptance Testing</i> pada Pembeli menurut skala Likert.....	137
Tabel 7.9 Hasil pengolahan data <i>User Acceptance Testing</i> pada Pembeli .....	137
Tabel 7.10 Hasil <i>User Acceptance Testing</i> pada Pihak Kantin.....	138
Tabel 7.11 Penilaian Bobot <i>User Acceptance Testing</i> pada Pihak Kantin menurut skala Likert.....	139
Tabel 7.12 Hasil pengolahan data <i>User Acceptance Testing</i> pada Pihak Kantin	139
Tabel 7.13 Pengujian Waktu pada Proses Bisnis Pemesanan dan Penjualan pada Kantin Filkom (to-be) .....	140



## DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Kantin Kantin Filkom UB.....	9
Gambar 2.2 Model Sistem Informasi .....	14
Gambar 2.3 Contoh diagram proses bisnis .....	20
Gambar 2.4 <i>Android Activity lifecycle</i> .....	31
Gambar 2.5 <i>Android Fragment lifecycle</i> .....	31
Gambar 2.6 Rational Unified Process Model .....	39
Gambar 3.1 Diagram Alur Penelitian .....	40
Gambar 4.1 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (as-is) 46	
Gambar 4.2 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (to-be) .....	49
Gambar 4.3 Bisnis Proses to-be Menyiapkan Penjualan .....	50
Gambar 4.4 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (to-be) selama masa transisi .....	51
Gambar 4.5 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom.....	57
Gambar 4.6 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Iterasi ke-1 Fase Elaborasi).....	58
Gambar 4.7 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Iterasi ke-1 Fase Konstruksi) .....	59
Gambar 5.1 Activity Diagram Edit Status Pesanan Selesai .....	70
Gambar 5.2 Activity Diagram Memesan Produk Penjualan.....	71
Gambar 5.3 Activity Diagram Membayar Pesanan.....	71
Gambar 5.4 Activity Diagram Mengisi Saldo E-Dompet .....	72
Gambar 5.5 Activity Diagram Mengelola Produk Penjualan .....	72
Gambar 5.6 Activity Diagram Tambah Produk .....	73
Gambar 5.7 Activity Diagram Menghapus Produk .....	73
Gambar 5.8 Activity Diagram Update Produk.....	73
Gambar 5.9 <i>Activity Diagram</i> Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi).....	74
Gambar 5.10 <i>Activity Diagram</i> Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi) .....	74
Gambar 5.11 <i>Sequence Diagram</i> Edit Status Pesanan.....	75

Gambar 5.12 <i>Sequence Diagram</i> Memesan Produk Penjualan .....	75
Gambar 5.13 <i>Sequence Diagram</i> Membayar Pesanan .....	76
Gambar 5.14 <i>Sequence Diagram</i> Mengisi Saldo E-Dompet.....	77
Gambar 5.15 <i>Sequence Diagram</i> Mengelola Produk Penjualan .....	78
Gambar 5.16 <i>Sequence Diagram</i> Tambah Produk.....	78
Gambar 5.17 <i>Sequence Diagram</i> Hapus Produk.....	78
Gambar 5.18 <i>Sequence Diagram</i> Update Produk .....	79
Gambar 5.19 <i>Sequence Diagram</i> Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi).....	79
Gambar 5.20 <i>Sequence Diagram</i> Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi) .....	80
Gambar 5.21 Class Diagram Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom.....	81
Gambar 5.22 <i>Physical Data Model</i> Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom.....	82
Gambar 5.23 <i>Physical Data Model</i> Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Awal Fase Construction).....	82
Gambar 5.24 Tampilan Pesanan .....	83
Gambar 5.25 Tampilan Kelola Produk .....	84
Gambar 5.26 Tampilan Tambah Produk .....	84
Gambar 5.27 Tampilan Hasil Penjualan .....	85
Gambar 5.28 Tampilan Menu Penjualan .....	85
Gambar 5.29 Tampilan Pembayaran .....	86
Gambar 5.30 Tampilan Riwayat Pemesanan .....	86
Gambar 5.31 Tampilan Profil Akun .....	87
Gambar 5.32 Tampilan Tambah Saldo .....	87
Gambar 6.1 Implementasi Basis Data Firebase (bank dan pesanan) .....	90
Gambar 6.2 Implementasi Basis Data Firebase (transaksi).....	91
Gambar 6.3 Implementasi Basis Data Firebase (user) .....	91
Gambar 6.4 Implementasi Antarmuka Pesanan .....	111
Gambar 6.5 Implementasi Antarmuka Pesanan (iterasi ke-1 fase konstruksi) ..	111
Gambar 6.6 Implementasi Antarmuka Kelola Produk .....	112

Gambar 6.7 Implementasi Antarmuka Kelola Produk (iterasi ke-1 fase konstruksi)	112
Gambar 6.8 Implementasi Antarmuka Tambah Produk	113
Gambar 6.9 Implementasi Antarmuka Update Produk	113
Gambar 6.10 Implementasi Antarmuka Hasil Penjualan	114
Gambar 6.11 Implementasi Antarmuka Atur Rentang Tanggal Hasil Penjualan	114
Gambar 6.12 Implementasi Antarmuka Menu Penjualan	115
Gambar 6.13 Implementasi Antarmuka Menu Penjualan (iterasi ke-2 fase konstruksi)	115
Gambar 6.14 Implementasi Antarmuka Menu Penjualan Telah Memesan	116
Gambar 6.15 Implementasi Antarmuka Menu Penjualan Telah Memesan (iterasi ke-2 fase konstruksi)	116
Gambar 6.16 Implementasi Antarmuka Pembayaran	117
Gambar 6.17 Implementasi Antarmuka Riwayat Pemesanan	117
Gambar 6.18 Implementasi Antarmuka Riwayat Pemesanan (iterasi ke-2 fase konstruksi)	118
Gambar 6.19 Implementasi Antarmuka Profil Akun	118
Gambar 6.20 Implementasi Antarmuka Tambah Saldo	119
Gambar 7.1 Hasil pengujian <i>Compatibility</i> menggunakan Test Lab Firebase	133

## DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	145
LAMPIRAN B HASIL UAT .....	147
B.1 UAT pada Pihak Kantin .....	147
B.2 UAT pada Pembeli .....	148



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Teknologi informasi pada masa globalisasi kini telah berkembang semakin cepat. Dengan perkembangan teknologi yang penuh dengan inovasi yang semakin canggih tersebut, gaya hidup pun berubah menjadi masyarakat yang semakin bergantung pada teknologi informasi untuk menjalankan berbagai kegiatannya. Pengembangan yang dapat mengoptimalkan teknologi tersebut dapat membuat masyarakat melakukan pengelolaan dan pencarian informasi secara lebih aktual, praktis dan optimal dengan menggunakan teknologi tersebut. Untuk mendapatkan pengembangan yang baik diperlukannya sebuah metode yang tepat agar pengembangan teknologi tersebut dapat benar-benar menjawab semua masalah.

Saat ini di banyak instansi telah menerapkan sistem informasi bahkan usaha usaha kecil seperti kantin telah mulai menerapkan sistem informasi didalamnya. Kantin adalah sebuah tempat dimana penjual dapat menjual makanan maupun minuman disebuah instansi perguruan tinggi. Kantin sendiri perlu menaati prosedur mengenai pengelolaan kantin. Biasanya para pembeli harus mengantri dalam sebuah jalur yang disediakan untuk membeli makanan. Dengan semakin banyaknya pengunjung dan keinginan pengunjung yang menginginkan pelayanan yang cepat dan tepat, makan perlunya pengembangan sistem informasi kantin yang dapat melayani pengunjung dengan cepat dan tepat.

Kantin Fakultas Ilmu Komputer (Filkom) sendiri merupakan tempat dimana mahasiswa filkom dapat membeli makanan dan minuman ketika tidak ada jam kuliah. Kantin filkom saat ini telah memiliki teknologi untuk menerapkan sistem informasi, namun hanya sebatas sistem informasi pada kasir kantin. Pada sistem informasi yang telah diterapkan saat ini kantin filkom masih belum dapat memberi pelayanan yang memuaskan terhadap pembeli. Terdapat beberapa permasalahan yang dapat mempengaruhi kepuasan pembeli yaitu, alur proses bisnis pemesanan yang cukup panjang. Saat ingin memesan makanan/minuman pembeli harus memindahtangankan kertas pesanan yang ditulis penjual ke kasir untuk pembayarannya lalu kembali ke penjual lagi lalu menunggu pesanannya digazebo. Masalah selanjutnya yakni penjual yang kebingungan dalam Mengantar Pesanan. Pada saat memesan makanan terkadang penjual menyuruh pembelinya untuk menunggu di gazebo filkom, namun pembeli tidak memberi posisi yang tepat dimana dia menunggu, hal itu yang membuat penjual kebingungan dalam mengantar pesanan. Maka dari itu diperlukannya suatu pengembangan sistem yang dapat membantu menangani masalah tersebut.

Untuk mengatasi masalah tersebut peneliti menggunakan modul pada *Customer Relationship Management* (CRM). CRM merupakan sebuah pendekatan baru dalam mengelola hubungan perusahaan dengan pelanggan pada level bisnis sehingga dapat memaksimalkan komunikasi, dan pemasaran melalui pengelolaan berbagai cara yang berbeda dengan pelanggan. Pendekatan ini memungkinkan

untuk mempertahankan pelanggan dan memberikan nilai tambah terus menerus pada pelanggan, selain itu juga memperoleh keuntungan yang berkelanjutan. Modul CRM yang akan diterapkan pada kantin filkom adalah pada operasional CRM *Sales force automation* (SFA) dan *Service Automation* (SA). Penerapan modul Sales Force Automation (SFA) adalah pada proses penjualan dan manajemen penjualan yang bertujuan memudahkan pelayanan penjualan atau proses pemesanan di Kantin Filkom UB, Sedangkan Service Automation (SA) adalah pada layanan berbasis *mobile/android* karena sistem yang dibangun untuk proses pemesanan yang tersedia di aplikasi pembeli sehingga pembeli dapat melayani dirinya sendiri. Untuk melakukan pengembangan sistem informasi, peneliti menggunakan Rational Unified Process (RUP). RUP merupakan kerangka kerja dalam pengembangan sistem informasi yang menggunakan teknik iteratif dan inkremental. Alasan menggunakan RUP karena dapat mengendalikan perubahan-perubahan yang terjadi pada sistem selama proses pengembangannya serta tahapan dalam RUP sangat jelas dan mudah digunakan.

Dalam penelitian sebelumnya yang berjudul “Pengembangan Sistem *Customer Relationship Management* (CRM) Self-Service Reservation Pada Everyday Smart Hotel Malang Menggunakan Usability Testing” yang ditulis oleh Farhan Ishami menyatakan bahwa dengan menggunakan *Customer Relationship Management* (CRM) Operasional modul *Sales Force Automation* (SFA) dan *Service Automation* (SA) aktivitas *Customer Self-Service* yang bertujuan untuk meningkatkan pelayanan dan kepuasan pelanggan dengan menerapkan sistem informasi berbasis web yang terintegrasi mendapatkan hasil keberhasilan 97% untuk pelanggan dan 92% untuk receptionist. Hasil waktu rata-rata pengerjaan didapat 97% untuk pelanggan dan 83% untuk receptionist lebih efisien. Kemudian pada aspek kepuasan pengguna memiliki enam belas pertanyaan masing-masing responden bernilai baik. Pada penelitian sebelumnya yang berjudul “Penerapan *Customer Relationship Management* (CRM) Sebagai Upaya Untuk Meningkatkan Loyalitas Pelanggan (Studi Kasus pada Bandung Sport Distro Malang)” yang ditulis oleh Anatasha Onna Carissa menyatakan bahwa dengan adanya otomatisasi layanan (*Service Automation*) yaitu melayani pembelian secara online dan adanya contact center untuk menangani keluhan pelanggan dapat meningkatkan loyalitas pelanggan pada Bandung Sport Distro Malang, dibuktikan dengan serangkaian proses yang telah dilakukan dan adanya tingkat pembelian kembali mencapai 25% dari total penjualan setiap bulan. Lalu pada penelitian yang dibuat oleh Agung Jasa Muttaqien Usmadi yang berjudul “Perancangan *Customer Relationship Management* Untuk Membangun Loyalitas Pelanggan Pada Divisi Pemasaran PT Bio Farma (Persero)” menyatakan bahwa E-CRM adalah solusi bagi PT Bio Farma (Persero) yang dapat memberikan ROI melalui otomatisasi pemasaran, layanan pelanggan, dan otomatisasi tenaga penjualan. Dari 2 penelitian tersebut dapat disimpulkan bahwa dengan menggunakan *Customer Relationship Management* (CRM) pada tataran Operasional khususnya pada *Sales Force Automation* (SFA) dan *Service Automation* (SA) sangat penting dalam meningkatkan loyalitas dan kepuasan pelanggan. Berdasarkan penelitian yang telah dilakukan Yuditia Fadhil Anisa dengan judul “Analisa Perancangan *Customer Relationship Management*



(CRM) Poin Pelanggan Pada Biro Komunikasi Pemasaran PT. Semen Indonesia (Persero) Tbk” menyatakan bahwa dengan menerapkan *Customer Relationship Management* (CRM) modul *Sales Force Automation* (SFA) dan *Service Automation* (SA), pada aspek *satisfaction* telah mempercepat proses poin yang sedang berjalan dan sistem dapat memudahkan toko mengetahui poin yang dimiliki. Untuk aspek pengguna toko sebanyak 33,3% setuju dan 66,7% sangat setuju akan efisiensi sistem.

Berdasarkan uraian latar belakang diatas, maka peneliti tertarik membuat pengembangan sistem informasi yang dapat mengatasi latar belakang tersebut. penelitian ini akan menggunakan modul *Customer Relationship Management* (CRM) sebagai metode pemecahan masalahnya yang akan diterapkan pada kantin Fakultas Ilmu Komputer Universitas Brawijaya, modul CRM tersebut adalah pada operasional CRM *Sales Force Automation* (SFA) dan *Service Automation* (SA). Lalu untuk melakukan pengembangan sistem informasi, peneliti menggunakan Rational Unified Process yang bertujuan untuk menghasilkan suatu sistem informasi yang dapat menjawab keinginan klien secara akurat dan dalam jangka waktu yang terukur. Judul penelitian yang diambil oleh peneliti yakni “Pengembangan Sistem Informasi *Sales Force Automation* Dan *Service Automation* Pemesanan Dan Penjualan Menggunakan Metode Rational Unified Process”. Melalui penelitian tersebut, peneliti berharap sistem informasi pemesanan dan penjualan kantin filkom dapat berguna untuk mempermudah proses pemesanan dan penjualan kantin filkom.

## 1.2 Rumusan masalah

1. Bagaimana hasil analisis proses bisnis dari sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode Rational Unified Process fase *inception*?
2. Bagaimana hasil analisis kebutuhan dan rancangan dari sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode Rational Unified Process fase *elaboration*?
3. Bagaimana hasil implementasi dan pengujian dari rancangan sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode Rational Unified Process fase *construction*?
4. Bagaimana hasil penerapan sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan dalam menangani permasalahan kantin FILKOM dengan menggunakan metode Rational Unified Process fase *trantition*?

### 1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

1. Menganalisis proses bisnis dari sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode *Rational Unified Process* fase *inception*.
2. Menganalisis kebutuhan dan Merancang sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode *Rational Unified Process* fase *elaboration*.
3. Melakukan implementasi dan pengujian hasil rancangan sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan pada Kantin FILKOM menggunakan metode *Rational Unified Process* fase *construction*.
4. Mengetahui hasil dari penerapan sistem informasi *Sales Force Automation* dan *Service Automation* pemesanan dan penjualan dalam menangani permasalahan kantin FILKOM dengan menggunakan metode *Rational Unified Process* fase *trantition*.

### 1.4 Manfaat

Manfaat yang ingin dicapai dari penelitian ini adalah mendapatkan hasil yang dapat mempermudah proses pemesanan dan penjualan pada kantin FILKOM.

### 1.5 Batasan masalah

Batasan masalah dari analisa ini adalah sebagai berikut:

1. Arsitektur sistem yang digunakan dalam pengembangan sistem ini berorientasi objek
2. Kerangka kerja pengembangan sistem informasi yang digunakan adalah Rational Unified Process.
3. Tahapan Rational Unified Process yang digunakan sampai pada fase *Transition*.
4. Sistem informasi yang dirancang adalah *Sales Force Automation* (SFA) dan *Service Automation* (SA) Pemesanan dan Penjualan pada Kantin Filkom.
5. Penelitian hanya dilakukan pada kantin FILKOM UB.
6. Sistem yang dibuat tidak sampai melibatkan pihak Bank sungguhan dan hanya melibatkan pihak Penjual dan Pembeli.

## 1.6 Sistematika pembahasan

### BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika pembahasan.

### BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang kajian-kajian pustaka yang digunakan untuk penelitian dan tentang pembahasan mengenai teori-teori yang mendasari penelitian.

### BAB III METODOLOGI

Bab ini menjelaskan langkah-langkah yang di tempuh atau metode penelitian yang digunakan dalam melakukan penelitian ini. Metode tersebut memuat dalam pengambilan data, dalam pengolahan data serta analisis dan hasil yang atau bahkan rekomendasi yang diberikan dari hasil penelitian ini.

### BAB IV ANALISIS KEBUTUHAN

Bab ini membahas pemodelan sistem informasi pemesanan dan penjualan yang akan dikembangkan pada kantin filkom, mulai dari pemodelan proses bisnis (as-is) dan pemodelan sistem (to-be).

### BAB V PERANCANGAN SISTEM

Bab ini membahas hasil perancangan sistem informasi pemesanan dan penjualan yang akan dikembangkan pada kantin filkom yaitu perancangan sequence diagram, class diagram, antarmuka dan perancangan database.

### BAB VI IMPLEMENTASI SISTEM

Bab ini membahas mengenai proses implementasi dalam membangun sebuah sistem informasi pemesanan dan penjualan kantin filkom yang sesuai dengan perancangan sistem yang telah dibuat.

### BAB VII PENGUJIAN SISTEM

Bab ini memuat proses hasil pengujian dan analisis sistem yang telah di implementasikan dan membandingkannya dengan rancangan dan tujuan awal sistem.

### BAB VIII PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Dalam kajian pustaka, penelitian ini dilakukan dengan mempelajari penelitian-penelitian terdahulu yang terkait dengan penelitian yang penulis teliti saat ini. Pada penelitian sebelumnya yang berjudul “Pengembangan Sistem *Customer Relationship Management* (CRM) Self-Service Reservation Pada Everyday Smart Hotel Malang Menggunakan *Usability Testing*” yang ditulis oleh Farhan Ishami menyatakan bahwa dengan menggunakan *Customer Relationship Management* (CRM) Operasional modul *Sales Force Automation* (SFA) dan *Service Automation* (SA) aktivitas *Customer Self-Service* yang bertujuan untuk meningkatkan pelayanan dan kepuasan pelanggan dengan menerapkan sistem informasi berbasis web yang terintegrasi mendapatkan hasil keberhasilan 97% untuk pelanggan dan 92% untuk receptionist. Hasil waktu rata-rata pengerjaan didapat 97% untuk pelanggan dan 83% untuk receptionist lebih efisien. Kemudian pada aspek kepuasan pengguna memiliki enam belas pertanyaan masing-masing responden bernilai baik.

Pada penelitian yang dibuat oleh Agung Jasa Muttaqien Usyadi yang berjudul “Perancangan *Customer Relationship Management* Untuk Membangun Loyalitas Pelanggan Pada Divisi Pemasaran PT Bio Farma (Persero)”. Menyatakan bahwa Hasil pengukuran loyalitas pelanggan pada PT Bio Farma (Persero) nihil, dikarenakan sistem yang selama ini dijalankan hanya berfokus pada sales in bukan sales out. Dengan kata lain, belum dapat membantu meningkatkan keuntungan dengan mengorganisir dan mengotomatisasi proses bisnis yang memelihara kepuasan dan loyalitas pelanggan. E-CRM adalah solusi bagi PT Bio Farma (Persero) yang dapat memberikan ROI melalui otomatisasi pemasaran, layanan pelanggan, dan otomatisasi tenaga penjualan.

Pada penelitian yang berjudul “Penerapan *Customer Relationship Management* (CRM) Sebagai Upaya Untuk Meningkatkan Loyalitas Pelanggan (Studi Kasus pada Bandung Sport Distro Malang)” yang ditulis oleh Anatasha Onna Carissa. Menyatakan bahwa dengan adanya otomatisasi layanan (*Service Automation*) yaitu melayani pembelian secara online dan adanya contact center untuk menangani keluhan pelanggan dapat meningkatkan loyalitas pelanggan pada Bandung Sport Distro Malang. Telah dibuktikan dengan serangkaian proses yang telah dilakukan dan adanya tingkat pembelian kembali mencapai 25% dari total penjualan setiap bulan.

Berdasarkan penelitian yang telah dilakukan S. Zafar Nasir dengan judul penelitian “*Sales Force Automation Using Mobile Technology*”. Penelitian ini tidak hanya berfokus pada otomatisasi saja, tetapi juga menambahkan atribut positif dalam efisiensi proses bisnis untuk mempengaruhi kekuatan penjualan dan proses bisnis lainnya. Penelitian ini telah mengidentifikasi banyak manfaat potensial seperti peningkatan efektivitas dan efisiensi penjualan, peningkatan produktivitas, dan peningkatan manajemen hubungan pelanggan.

Tabel 2.1 Rangkuman Kajian Pustaka

No.	Nama Penulis, Nama Jurnal, Tahun	Tujuan Penelitian	Metode Penelitian /Pengembangan	Hasil dan Kesimpulan
1	Agung Jasa Muttaqien Usmadi, Yeffry Handoko, Jurnal Tata Kelola dan Kerangka Kerja Teknologi Informasi, Vol. 2, No. 1, “Perancangan <i>Customer Relationship Management</i> Untuk Membangun Loyalitas Pelanggan Pada Divisi Pemasaran PT Bio Farma (Persero)”, Tahun 2016	Merancang <i>Customer Relationship Management</i> pada Divisi Pemasaran PT Bio Farma (Persero).	Framework Dynamic CRM	E-CRM adalah solusi bagi PT Bio Farma (Persero) yang dapat memberikan ROI melalui otomatisasi pemasaran, layanan pelanggan, dan otomatisasi tenaga penjualan.
2	Farhan Ishami, Retno Indah Rokhmawati, Mochammad Chandra Saputra, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 2, “Pengembangan Sistem <i>Self-Service Reservation</i> pada <i>Everyday Smart Hotel</i> Malang Menggunakan <i>Customer Relationship Management</i> (CRM) Operasional Modul <i>Sales Force Automation</i> dan <i>Service Automation</i> ”, Tahun 2018	Membuat sistem informasi <i>Self-Service Reservation</i> pada <i>Everyday Smart Hotel</i> Malang.	<i>Customer Relationship Management</i> (CRM) Evaluation	Dengan menggunakan <i>Customer Relationship Management</i> (CRM) Operasional modul <i>Sales Force Automation</i> (SFA) dan <i>Service Automation</i> (SA) aktivitas <i>Customer Self-Service</i> yang bertujuan untuk meningkatkan pelayanan dan kepuasan pelanggan dengan menerapkan sistem informasi berbasis web yang terintegrasi sangat baik.



No.	Nama Penulis, Nama Jurnal, Tahun	Tujuan Penelitian	Metode Penelitian /Pengembangan	Hasil dan Kesimpulan
3	Anatasha Onna Carissa, Achmad Fauzi, Srikandi Kumadji, Jurnal Administrasi Bisnis, Vol. 15, No. 1 “Analisa Perancangan <i>Customer Relationship Management</i> (CRM) Poin Pelanggan Pada Biro Komunikasi Pemasaran PT. Semen Indonesia (Persero) Tbk” Tahun 2014	Melakukan Analisis <i>Customer Relationship Management</i> (CRM) sebagai upaya untuk meningkatkan loyalitas pelanggan.	Analisis Data Non-Statistik	Dengan adanya otomatisasi layanan ( <i>Service Automation</i> ) yaitu melayani pembelian secara online dan adanya contact center untuk menangani keluhan pelanggan dapat meningkatkan loyalitas pelanggan pada Bandung Sport Distro Malang
4	S. Zafar Nasir, Syed Mujtaba Ahmed, 2012 8th International Conference on Computing and Networking Technology, “ <i>Sales Force Automation Using Mobile Technology</i> ” Tahun 2013	Melakukan analisis terhadap penggunaan <i>sales force automation</i> dalam perangkat bergerak	Analisis Data Non-Statistik	Penelitian ini telah mengidentifikasi banyak manfaat potensial seperti peningkatan efektivitas dan efisiensi penjualan, peningkatan produktivitas, dan peningkatan manajemen hubungan pelanggan.



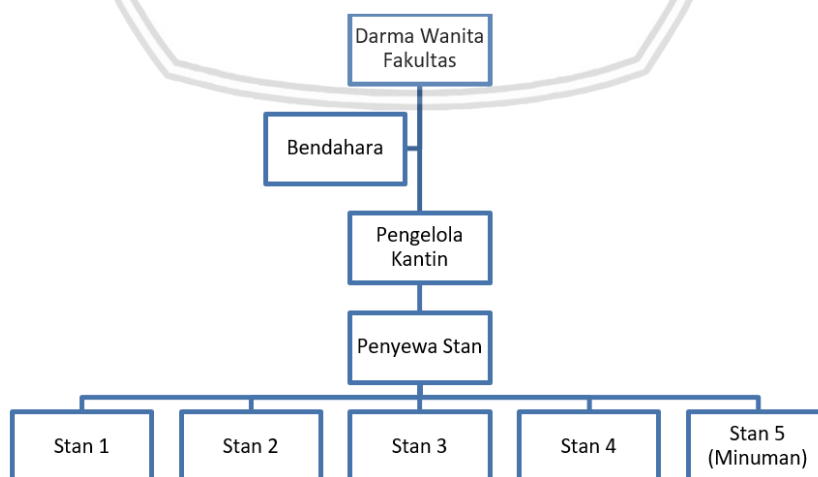
Berdasarkan Penelitian-penelitian diatas memiliki keterkaitan dengan penelitian yang dilakukan oleh penulis yaitu penelitian tersebut menggunakan CRM sebagai subjeknya. Dari 4 penelitian tersebut dapat disimpulkan bawah dengan menggunakan CRM pada tataran Operational khususnya pada *Sales Force Automation* (SFA) dan *Service Automation* (SA) sangat penting dalam meningkatkan loyalitas dan kepuasan pelanggan. Hal itu dapat membantu penulis untuk melakukan penelitian tentang pengembangan sistem informasi *Customer Relationship Management* pada tataran Operational khususnya pada *Sales Force Automation* (SFA) dan *Service Automation* (SA).

## 2.2 Kantin

Kantin merupakan ruangan dalam sebuah gedung umum yang dapat digunakan pengunjungnya untuk makan ataupun minum, baik yang dibawa sendiri maupun yang dibeli di sana. Kantin sendiri harus mengikuti prosedur tentang cara mengolah dan menjaga kebersihan kantin. Makanan dan minuman yang disediakan kantin haruslah bersih dan halal. Jenis-jenis yang disediakan pun minimal harus memenuhi 4 sehat 5 sempurna. Biasanya para pembeli harus mengantri dalam sebuah jalur yang disediakan untuk membeli makanan. Kantin hampir selalu ada di tiap sekolah dan sekolah tinggi di Indonesia. Biasanya kantin menjadi tempat berkumpul bagi para murid atau mahasiswa.

### 2.2.1 Kantin Fakultas Ilmu Komputer Universitas Brawijaya

Fakultas Ilmu Komputer Universitas Brawijaya (FILKOM UB) merupakan fakultas di universitas brawijaya yang tergolong maju, karena hampir semua difakultas Filkom diterapkan sistem informasi yang canggih, pada Kantin sendiri telah diterapkan sistem informasi untuk kasir yang dapat membantu pengelola dalam memantau kantin. Kantin Filkom merupakan kantin bagi mahasiswa Filkom untuk melakukan pemesanan makanan atau minuman. Struktur organisasi kantin Filkom dapat dilihat pada Gambar.



**Gambar 2.1 Struktur Organisasi Kantin Kantin Filkom UB**

Gambar 2.1 menjelaskan bahwa pemegang paling tinggi ialah darma wanita fakultas setelah setelah itu dipantau oleh Bendahara. Bendahara disini bertugas

memantau dan menyimpan keuangan darma wanita fakultas untuk nantinya digunakan dalam kegiatan-kegiatan darma wanita. Setelah itu dibawahnya ada pengelola kantin yang bertugas mengelola dan menangani operasional kantin. Dibawah pengelola kantin terdapat Penyewa Stan yang bertanggung jawab terhadap penyawaan pada stan-stan kantin, dan dibawah penyewa stan ada stan-stan penjual yang melakukan penjualan pada kantin Filkom.

## **2.3 Customer Relationship Management**

*Customer Relationship Management* (CRM), sudah di gunakan sejak awal 1990an. Sejak saat itu, telah banyak usaha untuk mendefinisikan domain CRM. Bahkan makna dari singkatan tiga huruf CRM ini diperebutkan. Sebagai contoh, walaupun kebanyakan orang akan mengerti bahwa CRM berarti *Customer Relationship Management*, yang lain menggunakan akronim untuk menggambarkan *Customer Relationship Marketing* (Buttle, 2009).

Perusahaan teknologi informasi cenderung menggunakan istilah CRM untuk menggambarkan aplikasi perangkat lunak yang digunakan untuk mendukung fungsi pemasaran, penjualan dan layanan bisnis. Ini sama dengan CRM dengan teknologi. Meskipun pasar untuk perangkat lunak CRM sekarang banyak dihuni oleh banyak pemain, komersialisasinya sangat meningkat pada tahun 1993 ketika Tom Siebel mendirikan Siebel Systems Inc. (sekarang merupakan bagian dari Oracle).

Kita dapat menyelesaikan perdebatan tersebut dengan memahami CRM sebagai tiga tipe, salah satunya adalah Operasional. Operasional CRM berfokus pada otomatisasi proses yang dihadapi pelanggan seperti penjualan, pemasaran dan layanan pelanggan.

### **2.3.1 Operasional CRM**

CRM operasional merupakan salah satu modul dari CRM. CRM operasional mengotomatiskan proses bisnis yang dihadapi pelanggan. Aplikasi perangkat lunak CRM memungkinkan fungsi pemasaran, penjualan dan layanan menjadi otomatis dan terintegrasi. Penelitian ini menerapkan modul Sales Force Automation (SFA) pada proses penjualan dan manajemen penjualan yang bertujuan memudahkan pelayanan penjualan atau proses pemesanan di Kantin Filkom UB, Sedangkan Service Automation (SA) pada layanan berbasis *mobile*/android karena sistem yang dibangun untuk proses pemesanan yang tersedia di aplikasi pembeli sehingga pembeli dapat melayani dirinya sendiri.

#### **2.3.1.1 Sales Force Automation**

*Sales Force Automation* (SFA) adalah bentuk asli dari CRM operasional. Sistem SFA sekarang banyak digunakan di lingkungan business-to-business dan dipandang sebagai 'keharusan kompetitif' yang menawarkan 'paritas kompetitif'.

*Sales Force Automation* (SFA) atau otomatisasi tenaga penjualan adalah aplikasi teknologi komputerisasi untuk mendukung staf penjualan dan manajemen penjualan dalam pencapaian tujuan yang terkait dengan pekerjaan mereka.

Sehingga pelanggan dapat langsung melaporkan dan memesan produk melalui aplikasi dengan menggunakan perangkat elektronik seperti komputer atau perangkat bergerak. Keuntungan yang dimiliki *Sales Force Automation* (SFA) adalah pelanggan dapat mengetahui aktivitas pemasaran, penjualan dan layanan terhadap pelanggan secara bersama (Payne, 2005). Sasaran yang terdapat pada *Sales Force Automation* (SFA) adalah (Buttle, 2009):

#### **A. Proses Penjualan / Manajemen Aktifitas**

Proses ini termasuk urutan aktifitas penjualan. Dimana arahan kepada sales representation yang terus menerus disetiap tahap yang berlainan dalam proses penjualan.

#### **B. Manajemen Penjualan**

Merupakan sarana yang mungkin bagi manajer perusahaan dan eksekutif dalam mengakses kebutuhan untuk aktifitas penjualan sebelum, selama, dan setelah order. Sarana ini memungkinkan manajer untuk mengatur tim yang berhubungan pada account pribadi, region dan industri.

#### **C. Manajemen Pesanan**

Manajemen pesanan memungkinkan perwakilan untuk mengubah kuota dan perkiraan menjadi pesanan dengan harga yang benar begitu pelanggan setuju untuk membeli. Jika ini dilakukan di depan pelanggan, pesanan dapat dimuat ke dalam produksi atau diambil dari gudang lebih cepat. Perangkat lunak manajemen pesanan mungkin termasuk mesin kutipan, modul penetapan harga dan konfigurator produk. Dengan visibilitas melalui portal, pelanggan, perwakilan, dan manajer memiliki akses ke informasi pesanan yang sama, terbaru, dan terkini.

#### **D. Konfigurasi Produk**

Konfigurasi produk memungkinkan penjual untuk merancang dan menetapkan harga produk, layanan, atau solusi yang disesuaikan. Konfigurator berguna ketika produk sangat kompleks atau ketika penyesuaian adalah bagian penting dari proposisi nilai.

#### **E. Ensiklopedia Produk**

Sebuah ensiklopedia produk adalah katalog produk elektronik yang dapat dicari yang umumnya berisi nama produk, nomor stok, gambar dan kation tertentu. Dapat disimpan di komputer dan / atau disediakan untuk pelanggan secara online.

#### **F. Visualisasi produk**

Visualisasi produk memungkinkan perwakilan penjualan untuk menghasilkan gambar produk yang realistis sebelum diproduksi. Visualisasi produk adalah aplikasi yang berguna ketika terhubung ke konfigurator

produk. Gambar dapat mengambil bentuk foto simulasi, model tiga dimensi atau gambar teknis, dan dapat menyertakan dokumentasi terkait lainnya seperti spesifikasi atau harga.

### 2.3.1.2 *Service Automation*

*Service Automation* (SA) atau otomatisasi layanan memungkinkan perusahaan untuk mengotomatisasi fungsi layanan perusahaan terhadap pelanggan melalui *call center*, *contact person*, *helpdesk*, layanan berbasis web atau bertatap muka langsung dengan pelanggan di lapangan. *Service Automation* (SA) juga menerapkan teknologi informasi internet khususnya web, melalui web suatu perusahaan dapat memberikan pelayanan sesuai keinginan pelanggan. Manfaat *Service Automation* (SA) bagi perusahaan dan pelanggan yaitu pelayanan kepada pelanggan lebih efektif dan efisien, mengurangi biaya layanan, meningkatkan kualitas layanan, dan meningkatkan produktivitas layanan ke pelanggan.

Adapun macam-macam aktivitas pada modul *Service Automation* (SA) yaitu (Buttle, 2009):

#### A. *Contact centre*

Media atau sarana bagi perusahaan untuk berkomunikasi dengan pelanggan. Operator *contact centers* bisa mengakses dengan siapa saja dia telah berkomunikasi dan apa saja isi dari percakapan yang telah dilakukan terkait dengan servis apa saja yang telah diberikan kepada pelanggan. Dengan adanya kemajuan teknologi saat ini juga memungkinkan untuk mengintegrasikan pusat komunikasi dengan divisi lain yang terkait untuk memudahkan operator dalam memberikan informasi kepada divisi lain yang terkait. Contoh dengan adanya aplikasi *komplain* berbasis elektronik, pelanggan bisa dengan mudah untuk menyampaikan keluhan yang terjadi di lapangan. Ketika pelanggan memasukkan jenis dan *komplain* apa yang dialami, maka sistem bisa dengan mudah memberikan informasi kepada divisi yang terkait untuk segera menanggapi dan menyelesaikan *komplain* tersebut.

#### B. *Call centres*

Sebagai pusat komunikasi dengan pelanggan via suara. Bisa dengan menggunakan jaringan telepon, jaringan *telpon seluler* maupun dengan bantuan teknologi berupa *VoIP*. Pada bagian ini dibutuhkan juga operator yang memiliki kemampuan lebih pada mendengar, dan juga kemampuan berkomunikasi yang baik. Peranan SA dalam *call center* dengan menggunakan *VoIP* bisa dimanfaatkan sebagai media berkomunikasi via internet dengan memberikan balasan secara otomatis sesuai dengan apa yang dibutuhkan oleh pelanggan.

#### C. *Helpdesks*

*Help desk* diasosiasikan dengan aplikasi IT dengan memberikan pelayanan langsung dengan perantara aplikasi seperti pelayanan

komplain, pusat informasi lowongan kerja dan pelayanan berbasis aplikasi bisa menggunakan konsep *helpdesk*.

#### D. Field service

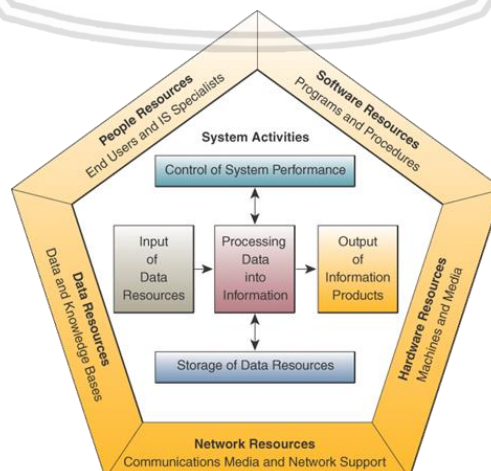
Pekerja teknis lapangan dalam jenis layanan B2C dan B2C banyak memanfaatkan layanan IT dalam menunjang pekerjaannya di lapangan. Data yang seringkali sangat dibutuhkan ketika berada di lapangan tidak seluruhnya bisa diakses bahkan dibawa. Dengan adanya teknologi, data tersebut bisa dikirim, diolah dan diakses kapan saja dan dimana saja. Contohnya untuk proses observasi di lapangan, data yang dibutuhkan bisa langsung diakses dari perangkat yang sudah terintegrasi, dan proses pendataan bisa langsung dilakukan dan diolah di lapangan tanpa harus membawa pulang datanya kembali ke kantor.

#### E. Customer self-service

Dengan semakin berkembangnya teknologi juga bisa memungkinkan bagi pelanggan untuk melakukan pemesanan barang, proses pembayaran, pengecekan pelayanan yang sudah dilakukan perusahaan dan pemantauan kinerja perusahaan yang bisa dilakukan 24 jam non stop.

### 2.4 Sistem Informasi

Sistem merupakan seperangkat komponen yang saling terkait, dengan batasan yang jelas, bekerja sama untuk mencapai suatu set umum tujuan. Sedangkan sistem informasi ialah kombinasi dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, kebijakan dan prosedur yang menyimpan, mengambil, mengolah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi digunakan oleh orang untuk berkomunikasi satu sama lain menggunakan hardware sebagai perangkat fisik, software untuk pemrosesan informasi dan prosedur, jaringan sebagai saluran komunikasi dan data yang tersimpan sebagai sumber data (O'Brien & Marakas, 2013).





### Gambar 2.2 Model Sistem Informasi

Sumber: O'Brien & Marakas (2010)

Dalam pengembangan sistem informasi terdapat beberapa tahapan yaitu, analisis kebutuhan, perancangan, implementasi, dan pengujian.

## 2.5 Proses Bisnis

Proses bisnis merupakan sekumpulan aktivitas yang memerlukan satu atau lebih jenis input dan membuat output yang bernilai bagi pelanggan. Proses bisnis terdiri dari sekumpulan aktivitas yang dilakukan secara koordinasi untuk mewujudkan tujuan bisnis. Setiap proses bisnis diberlakukan oleh satu organisasi, tapi memungkinkan berinteraksi dengan proses bisnis yang dilakukan oleh organisasi lain (Weske, 2007).

Berikut ini adalah jenis aktivitas dalam proses bisnis (Weske, 2007):

1. Aktivitas manual adalah aktivitas yang tidak didukung oleh sistem informasi. Contoh dari aktivitas manual adalah mengirimkan parcel kepada rekan bisnis.
2. Aktivitas interaksi pengguna adalah aktivitas yang dilakukan oleh karyawan menggunakan sistem informasi. Aktivitas ini perlu didukung antarmuka sistem yang baik agar pekerjaan yang dilakukan dapat lebih efektif. Contoh dari aktivitas interaksi pengguna adalah memasukkan data asuransi menggunakan sistem.
3. Aktivitas sistem adalah aktivitas yang tidak melibatkan peran pengguna. Aktivitas sepenuhnya dilakukan oleh sistem. Contoh dari aktivitas sistem adalah mengecek saldo rekening bank.

Proses bisnis direpresentasikan dengan model proses bisnis. Model proses bisnis diekspresikan dalam sebuah diagram proses bisnis. Diagram proses bisnis merupakan diagram yang merepresentasikan aktivitas-aktivitas dan tugas-tugas dari suatu proses dan hubungan antar keduanya. Diagram proses bisnis digambarkan menggunakan notasi grafik tertentu. Contoh notasi yang dapat digunakan untuk memodelkan proses bisnis adalah UML Activity Diagram dan BPMN (Business Process Model and Notation) (Weske, 2007).

### 2.5.1 Business Process Model and Notation (BPMN)

BPMN merupakan notasi grafis yang digunakan untuk memodelkan suatu proses bisnis. Tujuan utama dari BPMN ialah untuk memberikan notasi yang mudah dipahami oleh semua business users, mulai dari analis bisnis yang membuat konsep awal dari proses - proses, pengembang teknis yang bertanggung jawab dalam implementasi teknologi yang akan melakukan proses - proses tersebut, dan pelaku bisnis yang akan mengelola dan memantau proses-proses tersebut (Silver, 2012).

BPMN memiliki empat kategori elemen dasar yaitu (Silver, 2012):



### 1. *Flow object*.

*Flow object* adalah elemen utama yang menentukan perilaku proses bisnis. *Flow Object* terdiri dari events, aktivitas, gateways.

### 2. *Artefacts*.

*Artefacts* digunakan untuk menampilkan informasi tambahan tentang proses bisnis. *Artefacts* terdiri dari data object, groups, dan text annotations. Setiap artefacts dapat dikaitkan dengan flow elements.





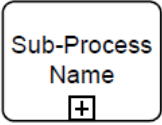
### 3. *Connecting object*.

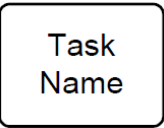





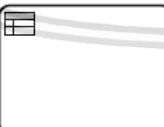

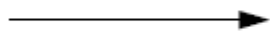
*Connecting object* menghubungkan flow object, swimlanes, atau artefacts. *Connecting object* terdiri dari sequence flows, message flows, association, dan data associations.


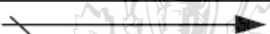



### 4. *Swimlanes*.

*Swimlanes* terdiri dari *pools* dan *lanes*. *Pools* merepresentasikan organisasi yang berpartisipasi dalam interaksi beberapa proses bisnis, masing-masing *pools* dilakukan oleh satu organisasi. *Lanes* merepresentasikan entitas organisasi, seperti departemen yang ada dalam organisasi tersebut.







**Tabel 2.2 Notasi BPMN**

Nama	Notasi	Deskripsi
Start Event		Digunakan untuk memulai proses
End Events		Digunakan untuk mengakhiri proses.
Intermediate Events		Menunjukkan suatu kejadian (Event) yang ada di antara awal dan akhir proses. Intermediate event hanya mempengaruhi aliran proses
Activity		Notasi ini merupakan kegiatan / aktivitas yang dilakukan perusahaan dalam sebuah proses.
Collapsed Sub-Process		Notasi ini merupakan aktivitas dimana didalam aktivitas tersebut terdapat proses yang terjadi.

Nama	Notasi	Deskripsi
Task		Task adalah bentuk atomic dari Activity. Task digunakan ketika pekerjaan dalam proses tidak dapat dipecah lagi ke tingkat yang lebih detail.
Service Task		Task yang menggunakan layanan otomatis, dapat berupa web service atau aplikasi.
Send Task		Task ini digunakan untuk mengirim pesan pada pihak eksternal dan selesai ketika pesan telah dikirim.
Receive Task		Task ini digunakan untuk menunggu dan menerima pesan dari pihak eksternal. Task ini selesai ketika pesan telah diterima.
User Task		Task ini menunjukkan bahwa tugas yang dilakukan melibatkan interaksi antara manusia dengan aplikasi.
Manual Task		Task ini menunjukkan bahwa tugas dilakukan tanpa bantuan alat proses bisnis atau aplikasi.
Business Rule Task		Task ini menyediakan mekanisme proses untuk memberikan masukan ke Business Rule Engine dan mendapatkan output yang diberikan oleh Business Rule Engine.
Script Task		Task yang menunjukkan tugas dilakukan oleh sebuah mesin proses bisnis dengan menjalankan suatu script.
Sequence Flow		Digunakan untuk menunjukkan urutan Flow

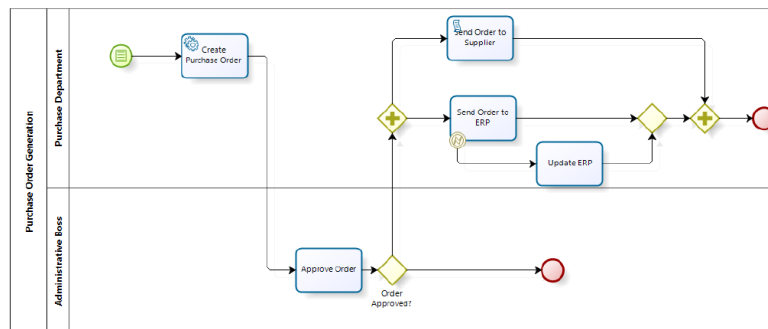
Nama	Notasi	Deskripsi
		Elements dalam sebuah proses dan hanya memiliki satu sumber dan satu target. Flow Elements terdiri dari: Events (Start, Intermediate, dan End), Aktivitas (Task dan Sub-Process), Choreography Activities (Choreography Task dan Sub-Choreography; untuk Choreographies), dan Gateways.
Conditional Flow		Sequence flow yang digunakan ketika ada sebuah kondisi, aliran akan diteruskan hanya jika kondisi bernilai true.
Default Flow		Notasi ini digunakan hanya ketika semua conditional flow bernilai false/not true pada saat runtime.
Message Flow		Digunakan untuk menunjukkan aliran pengiriman pesan antara dua partisipan.
Association / Asosiasi		Notasi yang digunakan untuk menghubungkan informasi dan artifact dengan flow object.
Gateway		Gateway ini menunjukkan pemecahan (karna adanya kondisi) atau penggabungan sequence flow dalam sebuah proses.

Nama	Notasi	Deskripsi
Gateway Control Types	<p>Exclusive  or </p> <p>Event-Based  </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>	<ul style="list-style-type: none"> <li>Exclusive (Keputusan): digunakan untuk membuat jalur alternatif dalam sebuah aliran proses.</li> <li>Event-based</li> <li>Inclusive: digunakan untuk membuat jalur alternatif atau paralel dalam sebuah aliran proses, semua ekspresi kondisi dievaluasi.</li> <li>Complex: digunakan untuk sinkronisasi jalur kompleks.</li> <li>Parallel: digunakan untuk menggabungkan dan membuat aliran paralel.</li> </ul>
Pool		Pool adalah representasi grafis dari partisipan. Partisipan dapat berupa sebuah perusahaan/organisasi atau mewakili peran yang lebih umum (misalnya, pembeli, penjual, atau produsen).
Message		Digunakan untuk menunjukkan isi dari komunikasi yang dilakukan antar dua partisipan.
Lane		Lane adalah sebuah bagian dalam proses atau pool yang panjangnya sama dengan panjang proses baik secara vertikal maupun horizontal. Lane digunakan untuk mengatur dan mengkategorikan aktivitas dalam sebuah pool. Lane terkadang digunakan untuk

Nama	Notasi	Deskripsi
		mewakili peran internal perusahaan (misalnya, manager, finance, shipping)
Data object	<p>Data Object</p>  <p>Data Objec (Collection)</p>  <p>Data Input    Data Output</p>  	<ul style="list-style-type: none"> <li>• Data Object: menunjukkan sebuah informasi yang ada pada sebuah proses seperti surat, dokumen bisnis, e-mail.</li> <li>• Collection: menunjukkan sekumpulan informasi.</li> <li>• Data Input-Output: Data input adalah masukan untuk seluruh proses, data ouput adalah data hasil dari seluruh proses.</li> </ul>
Group		Digunakan untuk mengelompokkan elemen grafis yang mempunyai kategori yang sama. Nama kategori muncul dalam diagram sebagai label grup.
Text Annotation	 <p>Descriptive Text Here</p>	Notasi ini digunakan untuk memberikan informasi teks tambahan dalam sebuah proses. Text Annotation dihubungkan menggunakan Asosiasi.

Sumber: Object Management Group (OMG) (2011)

Contoh diagram proses bisnis menggunakan BPMN:



**Gambar 2.3 Contoh diagram proses bisnis**

Sumber: BPMN by example (2014)

## 2.6 Analisis Kebutuhan Sistem Informasi

Rekayasa kebutuhan atau *Requirement Engineering* adalah nama lain dari analisis kebutuhan dimana setiap proses pengembangan perangkat lunak pasti melalui fase analisis kebutuhan. Rekayasa kebutuhan itu sendiri adalah sebuah proses untuk membangun layanan atau produk yang dibutuhkan oleh pelanggan atau pengguna beserta batasan-batasan yang diberlakukan atau aturan yang harus dipatuhi ketika produk atau layanan tersebut digunakan dan beroperasi serta ketika produk dan layanan tersebut sedang dibangun atau dikembangkan. Aktivitas-aktivitas dalam rekayasa kebutuhan dapat diwujudkan dengan cara-cara yang luas dan beragam namun memiliki tujuan untuk dapat memahami pemahaman apa yang dibutuhkan oleh pengguna atau pelanggan.

Kebutuhan pengguna secara umum dibagi menjadi dua kategori utama, yaitu Kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional menggambarkan apa yang sistem lakukan, sedangkan kebutuhan non-fungsional menggambarkan batasan-batasan yang harus dipenuhi oleh sistem dalam hal batasan kualitas dan batasan penggunaan (usability).

Secara umum, kebutuhan memiliki tiga tingkatan, yakni:

1. Kebutuhan normal: yaitu kebutuhan yang harus dipenuhi dan dinyatakan secara eksplisit oleh pengguna/stakeholder seperti fungsionalitas sistem dan performa Sistem.
2. Kebutuhan yang diharapkan (expected): yaitu kebutuhan yang tidak dinyatakan secara eksplisit namun menentukan kepuasan pelanggan, seperti: kemudahan interaksi dengan sistem, akurasi dan kebenaran proses.
3. Kebutuhan yang mengejutkan (exciting): yaitu kebutuhan yang melebihi kebutuhan normal untuk dapat lebih memuaskan *customer*, seperti fungsionalitas tambahan sistem lainnya yang lebih memuaskan *customer* ketika menggunakan sistem dalam menyelesaikan masalah, mencapai tujuan pengguna, dengan lebih efisien.

Aktivitas-aktivitas dalam rekayasa kebutuhan terdiri dari 4 proses utama, yaitu:

1. Penggalan dan analisis kebutuhan (software requirement elicitation and analysis). Dalam proses ini, developer harus bekerja bersama-sama dalam memahami domain aplikasi/sistem, layanan-layanan sistem yang harus disediakan, unjuk kerja sistem yang diperlukan, batasan-batasan perangkat keras dan sejenisnya.
2. Spesifikasi kebutuhan (software requirement specification). Proses untuk menjelaskan kebutuhan perangkat lunak yang telah didefinisikan sebelumnya secara lebih detail, tepat, dan terukur, dimana pernyataan



spesifikasi kebutuhan yang didefinisikan dalam proses ini akan menjadi dasar bagi perancangan dan implementasi.

3. Validasi & verifikasi kebutuhan (software requirement validation and verification). Proses pemeriksaan untuk menjamin bahwa pernyataan kebutuhan yang telah didefinisikan dalam dokumen SRS telah dispesifikasikan dengan benar, akurat, dan lengkap.
4. Manajemen kebutuhan (software requirement management). Aktivitas untuk melakukan kendali terhadap kebutuhan yang sedang maupun telah didefinisikan jika terjadi perubahan ketika sistem sedang dibangun dan dikembangkan serta setelah sistem digunakan secara operasional.

Pernyataan kebutuhan yang baik adalah pernyataan kebutuhan yang SMART, dimana:

1. Specific. Dalam satu pernyataan menggambarkan hanya satu aspek dari kebutuhan atau kualitas dari Sistem.
2. Measurable. Dimana kualitas dari sistem yang didefinisikan dinyatakan secara objektif dan kuantitatif.
3. Achievable. Secara teknis dan biaya, kebutuhan yang didefinisikan mampu dicapai dan terjangkau.
4. Relevant. Kebutuhan yang dinyatakan harus sesuai di tingkat yang dispesifikasikan.
5. Traceable. Jika terdapat kebutuhan turunan (kebutuhan yang diperlukan untuk memenuhi kebutuhan lainnya) harus dengan jelas mengalir dari pernyataan kebutuhan induknya sehingga seluruh kebutuhan yang didefinisikan dapat ditelusuri dengan baik.

Dalam sebuah pernyataan kebutuhan, setiap pernyataan harus menyatakan siapa yang bertanggung jawab/terlibat (WHO) dan apa yang harus dilakukan (WHAT). Selain itu dalam sebuah pernyataan kebutuhan dapat mengekspresikan seberapa baik sesuatu yang dilakukan oleh sistem harus dipenuhi, atau mengemukakan batasan apa yang harus dipatuhi ketika sistem melakukan sesuatu. Pernyataan kebutuhan yang baik adalah pernyataan kebutuhan yang:

1. Bebas dari istilah-istilah yang ambigu.
2. Bebas dari kata-kata yang bersifat indefinitif, seperti: ini, itu, seperti ini, seperti itu, demikian.
3. Bebas dari istilah-istilah yang tidak dapat diverifikasi, seperti fleksibel, user-friendly, tahan-lama, kokoh, ringan, cukup, kecil, portabel, dengan mudah, dengan cepat, dengan baik, dan kata-kata sifat lainnya.
4. Bebas dari implementasi. Sebuah pernyataan tidak menyatakan bagaimana cara menyelesaikan suatu kebutuhan, tapi menyatakan apa

yang diperlukan. Nyatakan masalah yang perlu diselesaikan, bukan solusinya.

5. Perlu dan memang diperlukan oleh pengguna untuk menyelesaikan masalah atau mencapai tujuan pengguna.
6. Bebas dari deskripsi operasional.
7. Bebas dari hal-hal yang perlu didefinisikan lebih lanjut atau di lain waktu.

## 2.7 Perancangan Sistem Informasi

Perancangan adalah sebuah proses mendefinisikan sesuatu yang akan dikerjakan dengan teknik bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya. Perancangan perangkat lunak adalah proses dimana analisis diterjemahkan menjadi blueprint untuk membangun perangkat lunak. Awalnya, blueprint menggambarkan pandangan menyeluruh perangkat lunak, yaitu desain pada abstraksi tingkat tinggi yang dapat langsung ditelusuri pada sistem tertentu, objektif, data rinci, dan fungsionalitas. Namun perbaikan berikutnya mengarah pada representasi desain dengan tingkat abstraksi yang jauh lebih rendah (Pressman, 2010).

Berikut ini merupakan prinsip dalam melakukan perancangan (Pressman, 2010):

- Hasil rancangan harus dapat dilacak dalam model persyaratan.
- Arsitektur sistem yang akan dibangun selalu dipertimbangkan.
- Rancangan dari data dan rancangan fungsi memiliki urgensi yang sama.
- Harus hati-hati dalam merancang antarmuka.
- Desain antarmuka pengguna harus sesuai dengan kebutuhan pengguna dan berfokus pada kemudahan pengguna.
- Komponen harus memiliki keterikatan yang rendah antara satu sama lain dan dengan lingkungan eksternal.
- Model rancangan harus mudah dimengerti.
- Rancangan harus dikembangkan secara iteratif.

Pada tahap ini memungkinkan tim pengembang untuk menilai dan meningkatkan kualitas dari hasil perancangan, sebelum lanjut ke tahap selanjutnya. Model rancangan dinilai oleh tim pengembang sebagai upaya untuk menentukan apakah terdapat kesalahan, inkonsistensi, kelalaian, adakah alternatif yang lebih baik? Dan apakah model rancangan tersebut dapat dilaksanakan dalam jadwal, biaya, dan batasan yang telah ditetapkan.

## 2.8 Unified Modelling Language (UML)




Model merupakan bentuk penyederhanaan dari kenyataan yang menyediakan deskripsi sistem yang lengkap berdasarkan sudut pandang tertentu. Sebuah model dapat menggambarkan struktur dan perilaku sistem, menampilkan hubungan antar elemen sistem. Pemodelan dapat membantu untuk menjaga konsistensi antar artefak sistem dan meningkatkan kemampuan tim pengembang dalam menangani kompleksitas perangkat lunak (Booch, et al., 1998).

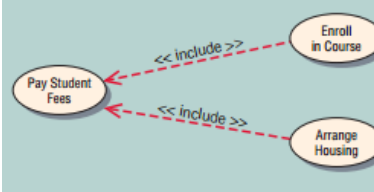
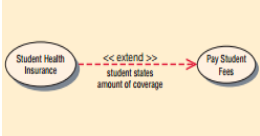
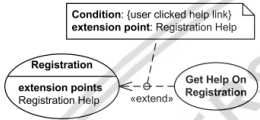
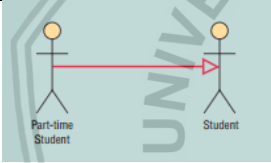
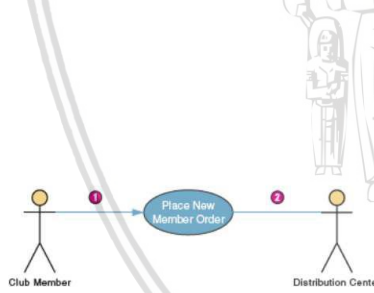
UML merupakan bahasa standar untuk memvisualisasikan hasil analisis dan desain berorientasi objek. UML menyediakan kumpulan standarisasi notasi dan diagram untuk membuat artefak dari pengembangan. Terdapat 13 jenis diagram yang termasuk pada UML, diantaranya adalah Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Use Case Diagram, Diagram Sequence, Communication Diagram, State Diagram, Activity Diagram, Deployment Diagram, Package Diagram, Timing Diagram, dan Interaction Overview Diagram (Booch, et al., 1998).

### 2.8.1.1 Diagram Use Case

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu dalam Use Case Specification. Namun, use case hanya menjelaskan tentang apa saja yang dilakukan oleh aktor dan sistem dan tidak menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case specification* terdiri dari nama use case, deskripsi singkat (Brief Description), Aliran Normal (Basic flow), Aliran Alternatif (Alternate Flow), Special Requirement, Pre-Condition dan Post-Condition. Simbol-simbol yang digunakan pada diagram use case dapat dilihat pada Tabel 2.3 (Kendall & Kendall, 2011).

Tabel 2.3 Simbol-simbol Diagram Use Case

Simbol	Deskripsi
	<i>Use case</i> mendeskripsikan fungsi dari sebuah sistem dilihat dari sudut pandang pengguna.
	<i>Actors</i> merupakan sesuatu yang berinteraksi dengan sistem untuk saling bertukar informasi. <i>Actors</i> tidak harus berupa manusia.
	<i>Communicates relationship</i> digunakan untuk menghubungkan aktor dengan use case.


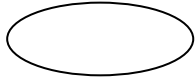

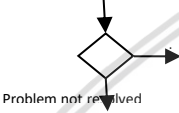
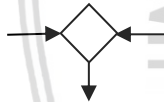
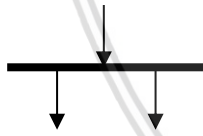
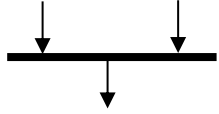
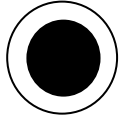
 <pre> graph LR     A((Pay Student Fees)) -.-&gt; &lt;&lt;include&gt;&gt;  B((Enroll in Course))     A -.-&gt; &lt;&lt;include&gt;&gt;  C((Arrange Housing))         </pre>	<p><i>Include relationship</i> atau disebut juga <i>uses relationship</i> menggambarkan situasi dalam suatu <i>use case</i> yang termasuk dalam <i>use case</i> lainnya.</p>
 <pre> graph LR     A((Student Health Insurance)) -.-&gt; &lt;&lt;extend&gt;&gt;  B((Pay Student Fees))     A -.-&gt; student states amount of coverage  B         </pre>  <pre> graph LR     A((Registration)) -.-&gt; &lt;&lt;extend&gt;&gt;  B((Get Help On Registration))     A -.-&gt; Condition: (user clicked help link)  B         </pre>	<p><i>Extends relationship</i> bertujuan untuk menggambarkan situasi dimana sebuah <i>use case</i> memiliki <i>behavior</i> yang memungkinkan <i>use case</i> lain dapat menggunakannya sebagai dasar variasi ataupun pengecualian.</p>
 <pre> graph LR     A[Part-time Student] --&gt; B[Student]         </pre>	<p><i>Generalizes relationship</i> menunjukkan bahwa satu hal lebih umum daripada hal lain.</p>
 <pre> graph LR     A[Club Member] -- 1 --&gt; B((Place New Member Order))     C[Distribution Centre] -- 2 --&gt; B         </pre>	<p><i>Associations</i> adalah sebuah relasi antara seorang <i>actor</i> dengan sebuah <i>use case</i> di mana terjadi interaksi antara mereka. Asosiasi dengan panah tertutup (1) di ujung yang menyentuh <i>use case</i> mengindikasikan bahwa <i>actor</i> di ujung yang satu lagi melakukan <i>use case</i> tersebut. Sedangkan asosiasi tanpa panah (2) mengindikasikan sebuah interaksi dari <i>use case</i> ke <i>actor</i> yang menerima hasil dari <i>use case</i> tersebut.</p>

Sumber: Kendall dan Kendall (2011)

### 2.8.1.2 Diagram Aktivitas

Diagram aktivitas adalah salah satu dari diagram UML yang digunakan untuk memodelkan aspek dinamis dari sistem. Diagram aktivitas pada umumnya merupakan sebuah flowchart yang menunjukkan aliran kontrol dari aktivitas ke aktivitas lainnya (Booch, et al., 1998). Tabel 2.3 merupakan keterangan dari simbol-simbol yang digunakan pada diagram aktivitas (Whitten & Bentley, 2007).

Tabel 2.4 Keterangan Simbol-Simbol Activity Diagram

Simbol	Deskripsi
	<i>Initial node</i> berupa lingkaran penuh yang menggambarkan titik mulai suatu proses.
	<i>Actions</i> adalah notasi yang menggambarkan langkah-langkah yang terjadi.
	<i>Flow</i> (alur) merupakan panah dalam diagram yang mengindikasikan alur antar- <i>actions</i> .
	<i>Decision</i> memiliki bentuk seperti wajik dengan satu alur masuk dan dua atau lebih alur keluar, alur keluar ditentukan dengan kondisi tertentu.
	<i>Merge</i> adalah wajik dengan dua atau lebih alur masuk dan satu alur keluar untuk menggabungkan alur yang sebelumnya terpisah oleh <i>decision</i> .
	<i>Fork</i> adalah bar hitam dengan satu alur masuk dan dua atau lebih alur keluar, aksi di bawah percabangan dapat terjadi dalam urutan apapun atau bahkan secara bersamaan.
	<i>Join</i> adalah bar hitam dengan dua atau lebih alur masuk dan satu alur keluar untuk menyatukan lagi alur aksi yang dipisahkan oleh <i>fork</i> .
	<i>Activity final</i> berbentuk lingkaran penuh dengan satu lingkaran di luarnya untuk menggambarkan titik akhir proses.





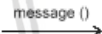
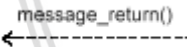


Sumber: Whitten dan Bentley (2007)

### 2.8.1.3 Diagram Sequence

Diagram Sequence merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. Pesan dapat berupa sinyal atau panggilan terhadap suatu operasi. Notasi pesan untuk panggilan terhadap suatu operasi dapat dituliskan dalam sintaks UML atau sintaks bahasa pemrograman tertentu.

Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 1998). Simbol-simbol yang digunakan pada diagram Sequence ditunjukkan pada Tabel 2.4 (Pressman, 2010).

**Tabel 2.5 Simbol-Simbol Diagram Sequence**

Simbol	Deskripsi
 atau 	<i>Entity</i> , entitas yang mempunyai atribut yang memiliki data yang bisa direkam.
	<i>Boundary</i> , menghubungkan user dengan sistem.
	<i>Control</i> , untuk mengontrol aktivitas-aktivitas yang dilakukan oleh sebuah kegiatan.
	<i>Message</i> , pengiriman pesan.
	<i>Return Values</i> , ditampilkan dengan garis panah terputus yang menggambarkan hasil dari pengiriman pesan. Digambarkan arah dari kanan ke kiri.
	Garis kehidupan ( <i>Lifelines</i> ), Garis 26vertical putus-putus yang memanjang kebawah dari simbol <i>actor</i> dan sistem yang mengindikasikan urutan kehidupan.
	<i>Bar</i> aktivasi, <i>Bar</i> didalam garis kehidupan ( <i>lifetime</i> ) yang menunjukkan periode waktu ketika peserta aktif dalam interaksi.

Sumber: Pressman (2010)

### 2.8.1.4 Diagram Kelas

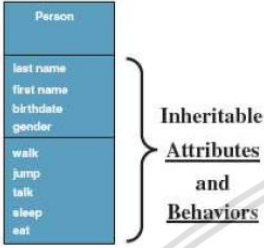
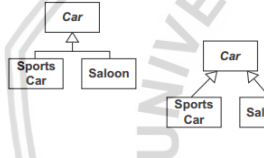


Class diagram atau diagram kelas digunakan untuk memodelkan tampilan statis perancangan dari sebuah sistem. Diagram kelas tidak hanya penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural, tapi juga penting untuk menyusun sistem yang dapat dijalankan melalui teknik forward maupun reverse. Diagram kelas secara umum memuat tiga hal, yaitu kelas,



interfaces, dan hubungan dependensi, generalisasi, serta asosiasi (Booch, et al., 1998).

Pada Tabel 2.5 menjelaskan tentang simbol-simbol yang digunakan dalam pembuatan class diagram pada umumnya (Whitten & Bentley, 2007).

**Tabel 2.6 Simbol-Simbol Class Diagram**

Simbol	Deskripsi
 <p>The diagram shows a class box for 'Person'. The top section is the class name. The middle section lists attributes: last name, first name, birthdate, and gender. The bottom section lists behaviors: walk, jump, talk, sleep, and eat. A bracket on the right side groups the attributes and behaviors under the label 'Inheritable Attributes and Behaviors'.</p>	<p><i>Attribute</i> adalah sekumpulan data yang dimiliki oleh objek.</p> <p><i>Behavior</i> adalah kumpulan dari sesuatu yang dapat dilakukan oleh objek dan terkait dengan fungsi-fungsi yang bertindak pada data objek (atribut). Pada siklus berorientasi objek, perilaku objek merujuk kepada metode, operasi, atau fungsi.</p>
 <p>The diagram shows two examples of inheritance. In the first, 'Sports Car' and 'Saloon' are subclasses of 'Car', indicated by solid lines with hollow triangle heads pointing to 'Car'. In the second, 'Sports Car' and 'Saloon' are subclasses of 'Car', indicated by solid lines with hollow triangle heads pointing to 'Car'.</p>	<p><i>Inheritance</i>, menunjukkan bahwa satu kelas merupakan turunan dari kelas lain.</p>
 <p>The diagram shows an association between 'Car' and 'Driver' classes, represented by a solid line connecting the two class boxes.</p>	<p><i>Association</i>, menunjukkan bahwa objek dari satu kelas berhubungan dengan kelas lain.</p>
 <p>The diagram shows an aggregation relationship between 'Car' and 'Engine' classes. A solid line connects the two class boxes, with an open diamond at the 'Car' end.</p>	<p><i>Agregation</i>, menunjukkan bahwa contoh objek dari satu kelas terdiri dari contoh objek dari kelas lain.</p>
 <p>The diagram shows a composition relationship between 'Car' and 'Body' classes. A solid line connects the two class boxes, with a filled diamond at the 'Car' end.</p>	<p><i>Composition</i>, menunjukkan hubungan dimana satu kelas bertanggung jawab atas pembuatan dan perusakan bagian-bagian dalam kelas lainnya. Jika satu kelas rusak, maka kelas lain juga rusak.</p>

Sumber: Whitten dan Bentley (2007)

## 2.9 Implementasi Sistem Informasi

Sistem telah dianalisis dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang sistem untuk diimplementasikan. Tahap ini merupakan implementasi dari tahap desain. Dalam tahap ini dilakukan pemrograman yang telah ditentukan, agar dapat dimengerti oleh komputer.

Pembuatan software dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum. Tahap implementasi sistem dapat terdiri dari langkah-langkah sebagai berikut ini:

1. Menerapkan rencana implementasi.
2. Melakukan kegiatan implementasi.
3. Tindak lanjut implementasi.

Konsep siklus hidup mempunyai implikasi bahwa setiap proyek pengembangan sistem harus dibagi dalam tahap-tahap berbeda dengan titik pengendalian manajemen yang formal diletakkan diantara tahap-tahap. Prinsip pengendalian dasar adalah setiap tahap harus menghasilkan dokumentasi secara formal ditelaah dan disetujui sebelum memulai tahap berikutnya dari siklus hidup proyek.

## 2.10 Android

Android merupakan sistem operasi yang diciptakan oleh Google yang dikhususkan untuk perangkat bergerak. Berbagai macam produsen telah menggunakan Android sebagai sistem operasi untuk perangkat yang mereka kembangkan. Android memiliki berbagai macam fitur yang sangat canggih seperti tampilan UI yang bagus baik dari segi user interface maupun user experience, dapat juga digunakan sebagai alat multimedia seperti pemutar musik dan video, dan juga menggunakan perangkat keras seperti *accelerometer*, *gyroscope*, dan sensor lainnya ke dalam aplikasi (Imaduddin & Permana, 2017).

Sejak dirilis pada tahun 2008, Android mengeluarkan beberapa versi dan hingga saat ini yang terbaru yaitu Oreo yang dirilis pada tahun 2017, bahkan saat ini sedang dikembangkan Android "P". Android menarik untuk para perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi sistem operasi untuk perangkat mereka. Dengan beberapa tersebut membuat perusahaan-perusahaan besar mengeluarkan produknya dengan menggunakan sistem operasi Android.

*Tool kit* yang biasa digunakan oleh para developer untuk mengembangkan aplikasi berbasis android yaitu *Android Software Development Kit* (SDK). Pada SDK terdapat beberapa tools untuk mengembangkan aplikasi seperti debugger, software libraries, emulator, dokumentasi, sample code, dan tutorial. Bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android adalah Java, namun ada beberapa Bahasa lainnya yang juga support untuk mengembangkan aplikasi Android seperti C++, serta Kotlin yang baru ditambahkan oleh Google sebagai bahasa yang support secara resmi untuk pengembangan Android.

Berbicara tentang pemrograman tentunya tidak lepas dengan *Integrated Development Environment* (IDE) yang bisa digunakan oleh para Developer. Pada tahun 2014 Google mengeluarkan IDE yang bernama Android Studio yang berbasis

IntelliJ IDE. Dengan menggunakan Android Studio para developer dapat membuat aplikasi dari nol hingga *publish* ke dalam *store*. Android Studio juga mempunyai beberapa fitur *built-in* yang sangat membantu para developer untuk memaksimalkan proses pembuatan aplikasi seperti Gradle, Code Completion, dan terintegrasi dengan beberapa services dari Google seperti Firebase.

### 2.10.1 Java

Java adalah sebuah bahasa pemrograman yang sangat populer yang dikembangkan oleh Sun Microsystems (saat ini dimiliki oleh Oracle). Dikembangkan lama setelah C dan C++, Java menggabungkan banyak fitur-fitur canggih dari bahasa-bahasa canggih tersebut, sambil mengatasi beberapa kelemahan mereka. Walaupun demikian, tingkat kecanggihan bahasa pemrograman bergantung pada library mereka. Library ini ada untuk membantu para developer untuk membuat aplikasi.

Java merupakan salah satu bahasa yang digunakan untuk mengembangkan Android. Selain java, ada beberapa bahasa lain yang bisa digunakan yaitu C/C++, dan pada pertengahan tahun 2017 Google resmi mendukung kotlin. Beberapa fitur inti Java adalah mudah dipelajari dan dimengerti, didesain untuk tidak bergantung kepada platform dan aman, menggunakan mesin virtual dan Bersifat object-oriented (fokus kepada objek program ketimbang logic). Android sangat bergantung kepada sifat-sifat dasar dari Java tersebut. Android SDK mengandung banyak library Java standar (library struktur data, library matematika, library grafik, library networking dan apapun yang dapat Anda inginkan) dan juga library khusus Android yang dapat membantu dalam mengembangkan aplikasi Android yang keren.

### 2.10.2 Android Studio

Android Studio adalah sebuah lingkungan pengembangan terintegrasi (*Integrated Development Environment* (IDE)) yang direkomendasikan oleh Google secara resmi untuk digunakan dalam pembangunan aplikasi perangkat bergerak pada platform Google Android setelah sebelumnya IDE pada platform Google Android menggunakan plugin *Android Developer Tools* (ADT) untuk Eclipse. Android Studio dirilis pada tanggal 16 Mei 2013 pada saat Google I/O 2013. Android Studio merupakan IDE yang dikembangkan berbasis IntelliJ IDEA, sebuah tools pembangunan aplikasi dan *code editor* berbasis IntelliJ yang mudah untuk digunakan dalam membangun aplikasi berbasis java.

Sejak bulan juni 2015 Google menganjurkan tidak menggunakan lagi Eclipse *Android Developer Tools* (Eclipse ADT) untuk membangun aplikasi Android. Secara efektif Google merilis update untuk ADT versi terakhirnya (versi 23.0.7) pada bulan Agustus tahun 2015 dan secara tidak langsung memaksa seluruh pengembang aplikasi android untuk beralih menggunakan Android Studio sebagai IDE-nya. Walaupun demikian untuk memudahkan para pengembang aplikasi android yang sebelumnya terbiasa menggunakan Eclipse dengan ADT ketika mengembangkan Aplikasi android, IDE Android Studio menyediakan fitur migrasi yang dapat

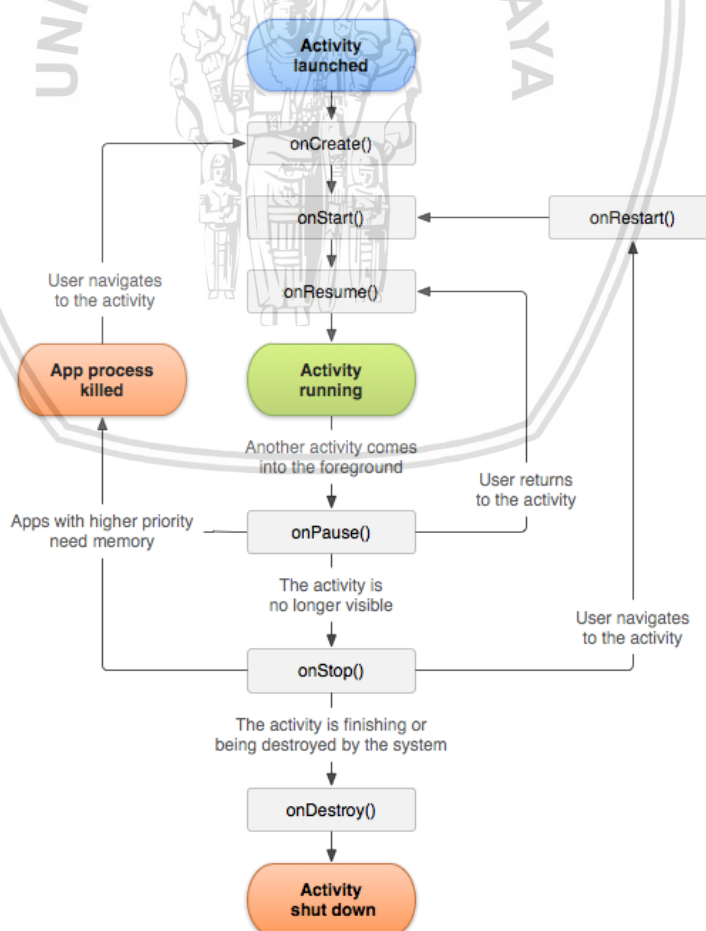
mempermudah proses konversi *source code project* aplikasi Android yang telah dibuat menggunakan Eclipse ADT untuk dapat dikembangkan lebih lanjut dengan menggunakan IDE Android Studio.

### 2.10.3 Fundamentals

Sebelumnya telah diketahui bahwa aplikasi android dibuat menggunakan bahasa java, dimana kode java dikompilasi bersama dengan file *resource* yang dibutuhkan oleh aplikasi. Semuanya di-*package* oleh tool yang dinamakan "*apt tools*", yang kemudian akan menghasilkan file apk. File apk tersebut nantinya akan diinstal ke android pengembang. Terdapat 6 fundamental atau komponen-komponen dasar pada sebuah aplikasi Android, yaitu (Imaduddin & Permana, 2017):

#### 2.10.3.1 Activity

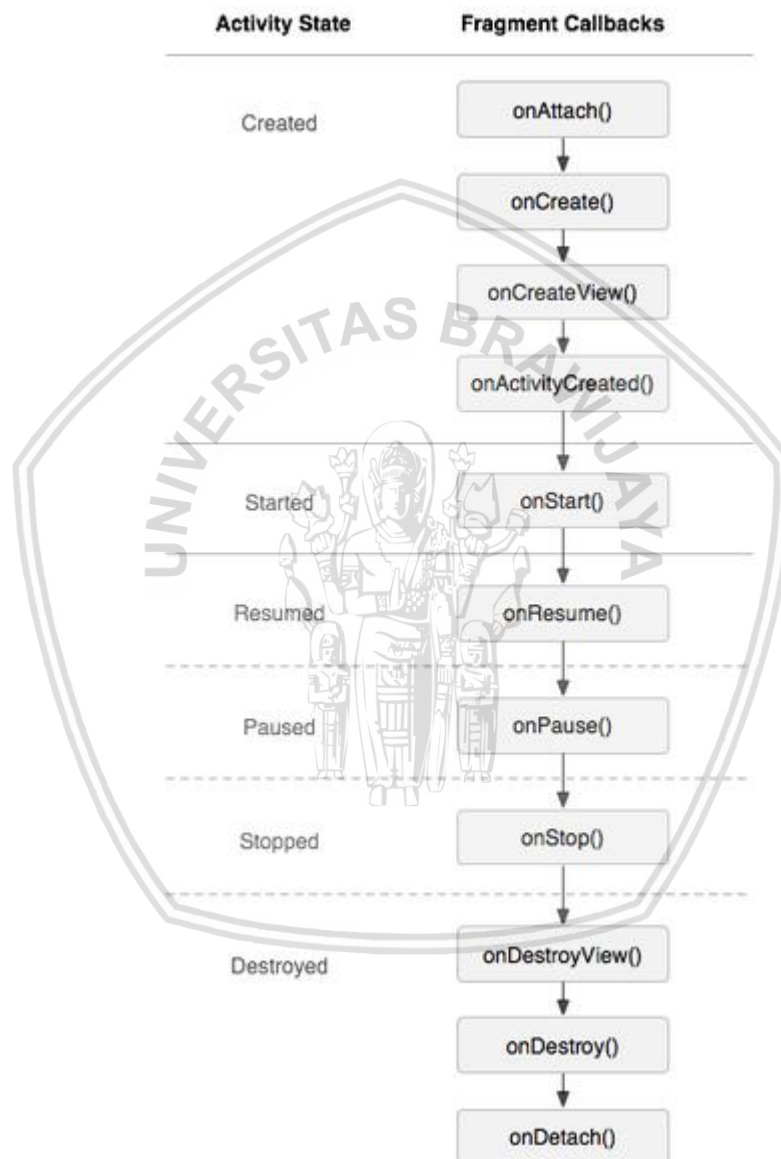
Merupakan satu komponen yang berhubungan dengan pengguna. *Activity* menangani tampilan mana yang akan ditampilkan ke dalam *User Interface (UI)*. Umumnya aplikasi android memiliki banyak activity, tetapi itupun tergantung kepada tujuan dan design aplikasi tersebut. Biasanya satu activity akan menampilkan satu UI. *Activity* memiliki *lifecycle* sendiri yang dimulai dari *onCreate* hingga *onDestroy*.



Gambar 2.4 Android Activity lifecycle

### 2.10.3.2 Fragment

Merupakan komponen *view* yang fleksibel dan dapat disisipkan pada sebuah tampilan dari *Activity*. *Fragment* memiliki daur hidup (*lifecycle*) sendiri dan bergantung pada *Activity* yang disisipkan. *Fragment* digunakan untuk memecah komponen pada aplikasi agar bisa lebih fleksibel dan mudah digunakan.



Gambar 2.5 Android Fragment lifecycle

Skema pada Gambar 2.4 menunjukkan method apa saja yang akan dipanggil didalam *fragment* ketika terjadi perubahan pada sebuah *activity*. Dari skema diatas menunjukkan bahwa perubahanan state dari *activity* akan mempengaruhi *lifecycle* dari sebuah *fragment*. Bisa dipastikan karena *fragment* merupakan komponen *view* yang bisa ditambahkan (embed) kedalam *activity*.



### 2.10.3.3 Intent

Komunikasi antar komponen didalam sebuah aplikasi merupakan hal yang sangat sering dilakukan, disinilah peran dari sebuah *Intent*. Fungsi dari *Intent* adalah dapat digunakan untuk menjalankan sebuah *activity*, mengirimkan pesan ke *BroadcastReceiver*, dan dapat juga digunakan untuk berkomunikasi dengan *Service* yang sedang berjalan. Ada tiga penggunaan umum *intent* dalam aplikasi Android yaitu:

1. Memindahkan satu **Activity** ke **Activity** lain dengan atau tidak membawa data.
2. Menjalankan *background Service* misal melakukan sinkronisasi ke server dan menjalankan proses berulang (*periodic/scheduler task*).
3. Mengirimkan obyek *broadcast* ke app yang membutuhkan. Misal jika aplikasi membutuhkan proses menjalankan sebuah *background Service* setiap aplikasi selesai melakukan *booting*. Aplikasi harus bisa menerima obyek *Broadcast* yang dikirimkan oleh sistem Android untuk *event booting* tersebut.

*Intent* memiliki dua bentuk yaitu:

1. **Explicit Intent** adalah tipe *intent* yang digunakan untuk menjalankan komponen aplikasi dengan tahu detail dari nama kelas yang dituju. Umumnya *intent* ini digunakan untuk mengaktifkan komponen pada aplikasi yang sama.
2. **Implicit Intent** adalah tipe *intent* yang tidak memerlukan detail nama kelas yang ingin diaktifkan. Ini memungkinkan komponen dari aplikasi lain bisa merespon *request intent* yang dijalankan. Pengguna *intent* ini umumnya diperuntukkan guna menjalankan fitur/fungsi dari komponen aplikasi lain.

### 2.10.3.4 Threads

Semua proses yang ada didalam sebuah aplikasi ada didalam *threads*. Secara *default* proses yang tampil dilayar merupakan proses yang ada didalam *MainThreads*, namu bukan hanya *MainThreads* saja yang bisa kita gunakan sebagai wadah untuk menjalankan proses dari sebuah aplikasi. Beberapa *threads* yang bisa digunakan adalah seperti *Handler* dan *AsyncTask*.

### 2.10.3.5 Service

*Service* merupakan komponen tidak terlihat yang dapat digunakan untuk menjalankan suatu proses yang memakan waktu lama atau yang membutuhkan komputasi intensif. Tidak seperti *Activity* dan *Fragment* yang memberikan pengalaman pada pengguna secara langsung. Komponen ini tidak memiliki antar muka dan bahkan pengguna tidak akan tahu bagaimana dia bekerja. Contohnya adalah pemutar musik dan *bloking opreration* untuk *networking*.

### 2.10.3.6 Receiver

*Receiver* atau *BroadcastReceiver* menggunakan *publish-subscribe* sebagai *design pattern*-nya. Ketika terjadi sebuah *event* (proses) kita bisa memberi tahu kepada komponen lain sehingga komponen lain dapat bereaksi terhadap *event* yang terjadi. Secara definisi, *BroadcastReceiver* adalah cara sederhana untuk merespon sebuah **BroadcastMessage** yang dipancarkan oleh dirinya sendiri,



aplikasi lain, atau sistem android. **BroadcastMessage** adalah pesan yang akan dipancarkan melalui obyek *intent*.

## 2.11 Firebase

Firebase adalah BaaS (*Backend as a Service*) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan *backend*. Ada banyak layanan yang diberikan oleh Firebase, layanan yang menarik adalah Firebase Real Time Database. Secara sederhananya, Firebase Real Time Database adalah layanan yang memberikan sebuah NoSQL database yang bisa diakses secara Real Time oleh pengguna aplikasi. Dan hebatnya adalah aplikasi bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan sync data segera setelah mendapatkan akses internet.

### 2.11.1 Firebase Real Time Database

Real Time Database adalah sebuah NoSQL database yang disediakan oleh Firebase. NoSQL database adalah database yang tidak menggunakan sistem relasi layaknya pada database tradisional (MySQL dll.). Metode penyimpanan data di dalam NoSQL menggunakan objek yang menggunakan format JSON (*JavaScript Object Notation*) (Firebase, 2017).

Firebase memungkinkan untuk menggunakan NoSQL database yang dibagikan kepada semua user, dan ketika terjadi perubahan data pada database tersebut, user akan segera mendapatkan update data secara real time. Tetapi bukan berarti database ini tidak mempunyai unsur keamanan, karena bisa diatur hak akses yang berbeda untuk setiap user. Salah satu fitur yang menarik adalah aplikasi bisa menyimpan data secara lokal ketika tidak ada koneksi internet, kemudian melakukan *sync* data segera setelah mendapatkan kembali koneksi internet. Real Time database ini cocok untuk aplikasi *instant messaging/chatting*.

## 2.12 Pengujian Sistem Informasi

Tahap ini merupakan tahap pengecekan dari software yang dibuat, apakah sudah sesuai dengan tujuan pembuatannya. Akan diuji oleh uji teknis secara fungsi (Black box testing) dan secara struktur program (White box testing) (Sommerville, 2011).

Tinjauan ahli dan uji coba terhadap produk. Pada tahap ini yang melakukan validasi atau tinjauan dari ahli isi atau materi dan ahli media yang bertujuan untuk mengetahui kelayakan dari produk yang dikembangkan. Apabila dalam tahap ini ada perbaikan terhadap media yang dikembangkan, maka masuk pada tahap revisi. Revisi bertujuan untuk menyempurnakan produk yang dikembangkan berdasarkan saran-saran yang diberikan. Setelah dianggap layak oleh ahli isi dan ahli media, kemudian dilakukan uji coba perorangan dan kelompok kecil. Pada ujicoba perorangan melibatkan 1 orang mahasiswa dan uji coba kelompok kecil melibatkan 10 orang mahasiswa dari fakultas ilmu komputer universitas

brawijaya. Apabila dalam ujicoba perorangan dan kelompok kecil ada perbaikan terhadap media yang dikembangkan, maka masuk pada tahap revisi. Revisi bertujuan untuk menyempurnakan produk yang dikembangkan berdasarkan saran saran yang diberikan.

### **2.12.1 Black-Box Testing**

Pengujian *black-box* atau juga bisa disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Yaitu, teknik pengujian *black-box* memungkinkan Anda untuk menentukan set kondisi input yang akan sepenuhnya menjalankan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif teknik *white-box*. Sebaliknya, itu adalah pendekatan komplementer yang kemungkinan akan mengungkap kelas kesalahan yang berbeda dari metode *white-box* (Pressman, 2010).

Pengujian *black-box* mencoba menemukan kesalahan dalam kategori berikut: (1) fungsi yang salah atau tidak ada, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses database eksternal, (4) kesalahan perilaku atau kinerja, dan, (5) inisialisasi dan terminasi kesalahan.

Tidak seperti pengujian *white-box*, yang dilakukan pada awal proses pengujian, pengujian *black-box* cenderung diterapkan selama tahap pengujian selanjutnya. Karena pengujian *black-box* dengan sengaja mengabaikan struktur kontrol, perhatian difokuskan pada domain informasi.

### **2.12.2 Compatibility testing**

Pengujian *Compatibility* merupakan adalah pengujian yang digunakan untuk memeriksa perangkat lunak yang dikembangkan mampu berjalan pada hardware, sistem operasi, aplikasi, ataupun lingkungan jaringan yang berbeda (Pressman, 2010). Tujuannya adalah untuk menemukan kesalahan yang khusus pada perangkat yang berbeda. Pengujian ini pada dasarnya dilakukan untuk mengetahui performa dari sistem pada sisi klien atau pengguna yang memiliki berbagai jenis cara untuk mengakses sistem tersebut. Semakin aplikasi dapat berjalan di banyak jenis perangkat yang berbeda, maka semakin baik aspek kompatibilitasnya.

Sifat *open source* yang ada pada Android, menjadikan banyaknya perangkat yang memakai Android sebagai sistem operasinya. Perangkat tersebut tentunya memiliki konfigurasi perangkat yang berbeda-beda seperti fitur, ukuran layar dan densitas layar. Fitur yang ada pada setiap perangkat bisa jadi berbeda antara satu dengan yang lainnya. Sehingga fitur yang digunakan untuk menjalankan fungsional aplikasi harus didefinisikan pada Android manifest, sehingga google play store dapat mendeteksi kompatibilitas aplikasi terhadap perangkat yang akan memasang aplikasi tersebut. Beragamnya versi Android juga harus diperhatikan agar aplikasi yang dikembangkan dapat berjalan di banyak versi. Maka perlu menetapkan minimum versi untuk menjalankan aplikasi. Konfigurasi layar yang berbeda-beda juga harus diperhatikan dalam pengembangan sehingga aplikasi dapat menjangkau banyak pengguna perangkat Android.

### 2.12.3 User Acceptance Testing

*User Acceptance Testing* (UAT) merupakan pengujian tingkat penerimaan sistem berdasarkan harapan user dari sistem tersebut. Pemangku kepentingan harus memastikan bahwa sistem yang telah dibangun dapat memenuhi kebutuhan dan harapan dari pengguna akhir agar sistem tersebut diterima atau digunakan. Pengujian UAT harus dapat diukur sehingga dapat diketahui sejauh mana sistem diterima oleh end user (Naik & Tripathy, 2008).

Pada pengujian *User Acceptance Testing* terdapat beberapa kriteria yang membantu dalam menilai penerimaan sistem. Namun dalam penerapannya tidak semua kriteria harus diuji pada sistem karena akan mengakibatkan terlalu banyak dan umumnya hasil pengujian. Tim pengembang dapat memilih bagian-bagian penting yang dapat mewakili penerimaan sistem. Berikut merupakan kriteria-kriteria yang ada pada pengujian UAT (Naik & Tripathy, 2008).

1. **Functional Correctness and Completeness**, Berfokus pada apakah sistem tersebut telah sesuai harapan dengan mengacu pada fitur yang telah didefinisikan.
2. **Accuracy**, Berfokus apakah perhitungan atau hasil dari sistem sesuai dengan nilai seharusnya.
3. **Data Integrity**, Berfokus pada performa sistem dalam mengolah data.
4. **Data Conversion**, Berfokus pada performa sistem dalam menjaga data pada saat terjadi perubahan jenis atau bentuk.
5. **Backup and Recovery**, Berfokus pada pencadangan dan pemulihan data
6. **Competitive Edge**, Berfokus pada keunggulan dibandingkan model yang lain.
7. **Usability**, Berfokus pada kemudahan dalam penggunaan sistem.
8. **Performance**, Berfokus pada kinerja sistem.
9. **Start-Up Time**, Berfokus pada waktu yang dibutuhkan untuk mulai beroperasi.
10. **Stress**, Berfokus pada kemampuan sistem menangani beban yang sangat tinggi atau stres.
11. **Reliability and Availability**, Berfokus pada keandalan perangkat lunak mengeksekusi tanpa kegagalan untuk jumlah waktu tertentu dalam lingkungan yang ditentukan.
12. **Maintainability and Serviceability**, Berfokus pada kemampuannya untuk menjalani perbaikan dan evolusi.
13. **Robustness**, Berfokus pada kemampuan sistem untuk pulih dari kesalahan, terus beroperasi dalam kondisi terburuk, dan beroperasi dengan andal untuk jangka waktu yang panjang.
14. **Timeliness**, Berfokus pada waktu yang dibutuhkan dalam *Time to market*.
15. **Confidentiality and Availability**, Berfokus pada kerahasiaan pada persyaratan bahwa data harus dilindungi dari pengungkapan yang tidak sah dan persyaratan bahwa data harus dilindungi dari penolakan layanan kepada pengguna yang berwenang.

16. **Compatibility and Interoperability**, Berfokus pada Kompatibilitas sistem atau sebagai kemampuan untuk beroperasi dengan cara yang sama di berbagai platform dan konfigurasi jaringan dan di berbagai campuran aplikasi lain.
17. **Compliance**, Berfokus pada kesesuaian sistem terhadap standar teknis yang relevan, seperti standar IEEE, standar antarmuka sistem operasi, dan standar IP.
18. **Installability and Upgradability**, Berfokus pada memastikan bahwa sistem dapat diinstal dengan benar dan ditingkatkan di lingkungan pelanggan.
19. **Scalability**, Berfokus pada kemampuan sistem untuk secara efektif memberikan kinerja yang dapat diterima.
20. **Documentation**, Berfokus pada tingkat kualitas panduan penggunaan sistem.

Dalam proses *User Acceptance Testing* terdapat beberapa tahap yang harus dilakukan (Sommerville, 2011) :

1. *Define acceptance criteria*

Tahapan awal dalam UAT adalah mendefinisikan kriteria apa yang akan dijadikan pedoman dalam pengujian.

2. *Plan acceptance testing*

Tahap perencanaan mengenai biaya, waktu dan partisipan dalam pengujian yang akan dilakukan.

3. *Derive acceptance tests*

Pada tahap ini terdapat perencanaan lebih mendetail mengenai pengujian sehingga pada akhirnya dapat mewakili pengujian pada seluruh kebutuhan fungsional dan non fungsional.

4. *Run acceptance tests*

Tahapan dimana pengujian dilakukan dan melibatkan calon pengguna. Idealnya sistem dijalankan dalam lingkungan sebenarnya namun hal tersebut mungkin dapat mengganggu atau membutuhkan sumber daya yang besar.

5. *Negotiate test results*

Tahapan dimana hasil dari pengujian diolah sedemikian rupa sehingga dapat diambil sebuah nilai besar penerimaan terhadap sistem.

6. *Reject/accept system*

Merupakan tahap akhir dimana pemangku kepentingan akan memberikan keputusan apakah sistem dapat dijalankan atau harus dilakukan perbaikan.

### 2.13 Skala Likert

Skala Likert merupakan salah satu cara yang berguna dalam mempelajari hubungan antara sikap dan atau persepsi individu diperiksa dan dianalisis. Pada implementasinya, skala ini digunakan dengan cara meminta beberapa responden untuk memberikan opini mengenai pertanyaan yang diajukan. Responden akan menilai sejauh mana mereka setuju ataupun tidak setuju mengenai solusi dari permasalahan yang ditentukan. Proses perhitungan pada skala likert diawali dengan memberikan bobot nilai pada setiap jawaban (Sugiyono, 2017).

**Tabel 2.7 Rincian Bobot Nilai**

A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Tidak Setuju	2
E = Sangat Tidak Setuju	1

Sumber: Sugiyono (2017)

Setelah bobot dan persentase nilai ditentukan, selanjutnya menghitung jumlah jawaban, dikalikan dengan bobot nilai. Berikut merupakan perhitungannya (Sugiyono, 2017):

1. Jawaban Sangat Setuju (A) =  $5n$
2. Jawaban Setuju (B) =  $4n$
3. Jawaban Netral (C) =  $3n$
4. Jawaban Tidak Setuju (D) =  $2n$
5. Jawaban Sangat Tidak Setuju (E) =  $1n$

$$TotalNilai = 5n + 4n + 3n + 2n + 1n \quad (2.1)$$

$n$  merupakan jumlah responden yang menjawab. Untuk melakukan perhitungan selanjutnya harus diketahui terlebih dahulu nilai terendah ( $Y$ ) dan nilai tertinggi ( $X$ ). Rumus yang digunakan dapat dilihat pada Rumus 2.2, dimana  $N1$  merupakan nilai tertinggi likert,  $n$  merupakan jumlah responden, dan  $U$  merupakan jumlah uji kasus.

$$Y = N1 \times n \times U \quad (2.2)$$

Sedangkan untuk menghitung hasil dari UAT dengan skala likert terdapat pada rumus 2.3.

$$Index\% = \left( \frac{TotalNilai}{Y} \right) \times 100 \quad (2.3)$$



## 2.14 Rational Unified Process

Rational Unified Process (RUP) adalah kerangka proses yang menyediakan simulasi sistem pada industri untuk sistem, software, implementasi, dan manajemen proyek yang efektif. RUP adalah salah satu dari sekian banyak proses yang terdapat di dalam Rational Process Library, yang memberikan simulasi terbaik untuk pengembangan atau kebutuhan proyek (IBM, 1998).

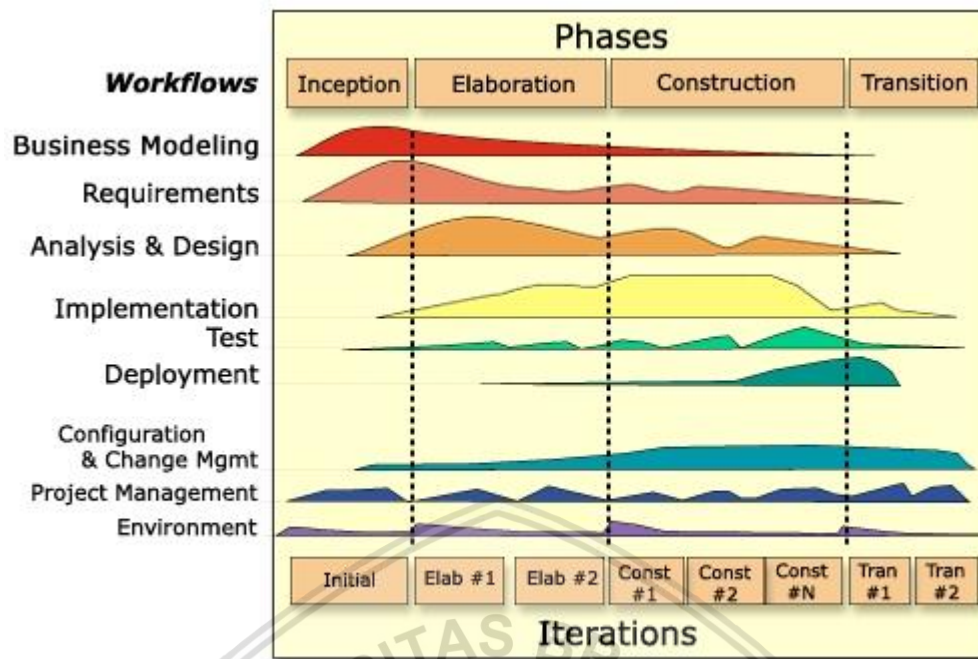
Tujuan utama dari penggunaan RUP ini adalah menciptakan suatu perangkat lunak yang dapat memenuhi kebutuhan klien dengan biaya yang telah ditentukan secara tepat waktu. Sebagai kerangka perangkat lunak unified process, RUP menerapkan teknik pengembangan perangkat lunak *iterative and incremental development*. Dengan dasar pengembangan perangkat lunak tersebut, RUP dapat menjadi suatu kerangka proses pengembangan perangkat lunak yang cukup dinamis dalam industri teknologi informasi.

RUP menggunakan konsep object oriented, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan Unified Model Language (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu major milestone yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas Inception, Elaboration, Construction, dan Transition.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration dan Change Manegement, Project Management, Environment (IBM, 1998).





Gambar 2.6 Rational Unified Process Model

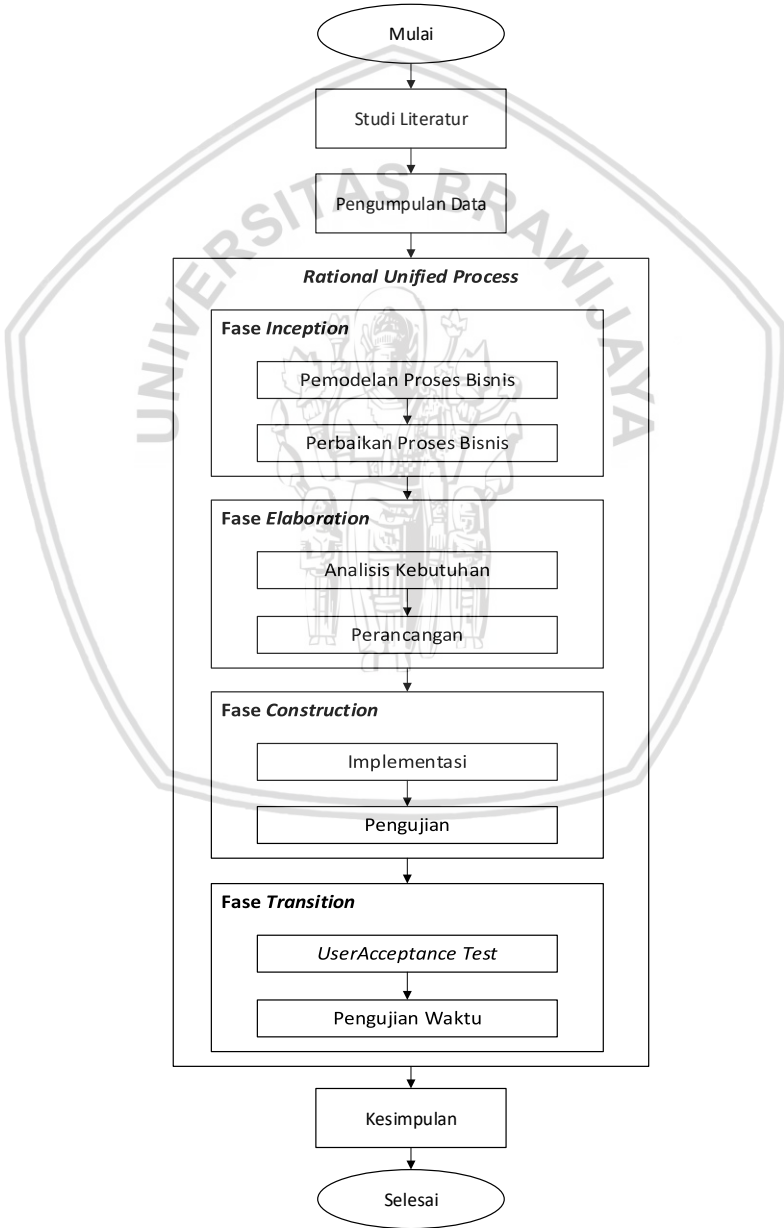
Sumber: IBM (1998)

Setiap tahapan pada siklus ini dapat dipecah-pecah menjadi beberapa iterasi. Iterasi-iterasi tersebut nantinya diharapkan akan menciptakan suatu produk akhir yang dapat memenuhi kebutuhan klien. Keuntungan dari proses iterasi ini adalah mengurangi resiko, dimana dengan membangun suatu produk secara langkah demi langkah maka resiko akan dapat di kenali lebih dini.

Untuk mengakomodir keperluan-keperluan dalam mengembangkan suatu sistem perangkat lunak, kerangka RUP ini terdiri dari beberapa tools yang terintegrasi yang dapat membantu pelaksanaan sebuah proyek.

### BAB 3 METODOLOGI

Penelitian ini merupakan penelitian implementatif. Dalam penelitian ini luaran produk yang dihasilkan adalah prototipe dari sistem. Pendekatan implementatif yang digunakan dalam penelitian ini adalah analisis dan perancangan sistem menggunakan pendekatan berorientasi objek. Langkah - langkah yang akan dilakukan dalam penelitian ini adalah studi literatur, pengumpulan data, pemodelan bisnis, perbaikan proses bisnis, analisis kebutuhan dan perancangan sistem, pengujian, UAT (*User Acceptance Test*), pengujian waktu dan yang terakhir yaitu kesimpulan. Gambar 3.1 menampilkan alur penelitian yang akan dilakukan dalam penelitian ini.



Gambar 3.1 Diagram Alur Penelitian

### 3.1 Studi Literatur

Studi literatur merupakan tahap mencari, mempelajari dan menyusun teori dasar dan referensi yang mendukung pengembangan sistem yang dibuat. Literatur yang digunakan berasal dari buku, jurnal atau paper, atau penelitian yang telah dilakukan oleh peneliti lain sebelumnya (kajian pustaka). Dalam penelitian ini dijelaskan teori-teori mengenai proses bisnis dan pemodelan proses bisnis menggunakan BPMN, sistem informasi, analisis dan perancangan berorientasi objek, UML sebagai bahasa pemodelan, serta evaluasi hasil analisis dan perancangan sistem. Sedangkan kajian pustaka pada penelitian ini yaitu kajian penelitian terdahulu tentang analisis dan perancangan menggunakan pendekatan berorientasi objek.

### 3.2 Pengumpulan Data

Teknik pengumpulan data dilakukan melalui wawancara dan observasi. Wawancara ini bertujuan untuk mendapatkan penjelasan mengenai proses bisnis kantin filkom dan informasi yang nantinya digunakan sebagai kebutuhan sistem. Sedangkan teknik observasi digunakan untuk melakukan pengamatan di lingkungan kantin untuk mendapatkan informasi-informasi yang dibutuhkan dalam penelitian ini. Berikut ini merupakan penjelasan lebih lengkap metode dalam melakukan pengumpulan data diantaranya:

#### 3.2.1 Wawancara

Wawancara dilakukan kepada pihak yang berwenang terhadap Kantin Fakultas Ilmu Komputer (Filkom) yaitu pihak pengelola Kantin untuk mengetahui proses bisnis pemesanan makanan dan minuman pada kantin filkom saat ini serta masalah yang dialami selama proses pemesanan tersebut. Data yang diperoleh wawancara berupa kebutuhan sistem yang diperlukan dan informasi proses bisnis pemesanan saat ini yang telah dilakukan oleh Kantin Filkom dan beberapa kendala yang diperoleh.

#### 3.2.2 Observasi

Observasi dilakukan untuk mengetahui proses pemesanan makanan dan minuman pada kantin filkom. Observasi dilakukan untuk mengetahui aktivitas proses pemesanan makanan dan minuman dari awal sampai akhir serta menganalisis *time to access* proses bisnis yang telah terjadi secara langsung mencari permasalahan yang dialami.

### 3.3 Fase *Inception*

Fase ini bertujuan untuk memodelkan sistem informasi pemesanan dan penjualan yang akan dikembangkan pada kantin filkom, mulai dari pemodelan proses bisnis (as-is) dan pemodelan sistem (to-be). Pada fase inception terdapat beberapa tahapan. Berikut ini merupakan penjelasan tahapan dalam fase *inception* diantaranya:

### 3.3.1 Pemodelan Proses Bisnis

Dalam melakukan pemodelan Proses Bisnis penulis harus melakukan observasi terlebih dahulu dan hasil dari wawancara untuk dapat melakukan pemodelan proses bisnis yang ada (*As-is*). Notasi grafis yang akan digunakan dalam memodelkan proses bisnis menggunakan *Business Process Model and Notation (BPMN)*. Setelah memodelkan proses bisnis akan dilakukan analisis permasalahan dari proses bisnis yang ada. Dengan adanya permasalahan yang ditemukan setelah memodelkan proses bisnis *as-is* penulis dapat dapat membuat rekomendasi yang tepat.

### 3.3.2 Perbaikan Proses Bisnis

Perbaikan Proses Bisnis yaitu memodelkan proses bisnis yang akan diusulkan (*To-be*) untuk memperbaiki/mengoptimalkan proses bisnis yang sebelumnya (*As-is*). Dari hasil analisis proses bisnis sebelumnya akan didapatkan sebuah solusi/rekomendasi yang nantinya akan di modelkan pada proses bisnis usulan (*to-be*). Proses bisnis usulan (*to-be*) ini akan menjadi dasar dalam mengembangkan sebuah sistem informasi pemesanan dan penjualan kantin filkom.

## 3.4 Fase *Elaboration*

Fase *elaboration* dilakukan setelah fase *inception*. Pada fase ini lebih pada analisis masalah dan menggambarkan/merancang alur sistem yang dibuat untuk pengerjaan pengembangan sistem agar mudah dikerjakan. Tujuan dari tahap elaborasi adalah untuk menganalisis masalah domain dan merancang arsitektur sistem.

### 3.4.1 Analisis Kebutuhan

Analisis kebutuhan ini bertujuan untuk mencari apa yang dibutuhkan, apa yang harus bisa dilakukan sistem, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional. Analisis ini dilakukan dengan menggunakan model RUP berdasarkan buku yang telah disusun oleh Kurt Bittner dan Ian Spence (Bittner & Spence, 2002). Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Melakukan analisis pemangku kepentingan dan pengguna yang meliputi analisis tipe dan peran pemangku kepentingan serta peran pengguna dari proses bisnis *to-be*.
2. Menganalisa kebutuhan pengguna dan pemangku kepentingan yang bertujuan untuk menjelaskan apa yang dibutuhkan oleh pengguna maupun pemangku kepentingan dari sistem yang akan diturunkan menjadi persyaratan sistem.
3. Menjelaskan kedudukan dari sistem yang akan dirancang.
4. Melakukan analisis untuk mendefinisikan fitur-fitur yang ada pada sistem yang akan dirancang. Analisis fitur ini menggunakan aturan MoSCoW untuk menentukan prioritas yang dimiliki masing-masing fitur.
5. Mengidentifikasi persyaratan produk yang mencakup kebutuhan fungsional dan non-fungsional dari sistem.

6. Melakukan pemodelan use case yang mencakup analisis aktor, diagram use case, dan spesifikasi use case.

### 3.4.2 Perancangan

Data-data pada tahap sebelumnya seperti kebutuhan *functional* dan *non-functional* yang telah didapat akan menjadi acuan untuk merancang diagram sistem. Tujuan pada tahap perancangan ini adalah untuk memodelkan sistem yang akan dibuat berdasarkan kebutuhan *functional* dan *non-functional* agar dapat diterapkan saat implementasi sistem. Pada tahap perancangan terdapat *activity diagram* yang bertujuan untuk memodelkan aktifitas user dengan sistem, *sequence diagram* yaitu diagram untuk memodelkan proses-proses dalam sebuah program, *class diagram* yang bertujuan untuk memudahkan programmer dalam membuat sebuah *class* pada sistem, *physical data model* yang bertujuan memodelkan data-data fisik/ basis data pada sebuah program dan yang terakhir *mockup user interface*.

### 3.5 Fase Construction

Fase *construction* dilakukan setelah fase *elaboration* telah dilakukan. Pada fase ini telah dilakukannya penulisan kode program hingga pengujian program tersebut. Fase *construction* terdapat 2 tahapan yaitu implementasi dan pengujian sistem. Berikut merupakan penjelasan tahapan dalam fase *construction* diantaranya:

#### 3.5.1 Implementasi

Implementasi merupakan suatu tahap penerjemahan atau penulisan kode program dan menggunakan beberapa pemodelan yang telah dibuat pada tahap perancangan sistem sebagai suatu referensi bagaimana suatu sistem tersebut akan di bentuk dan kebutuhan apa saja yang harus terpenuhi dari sistem tersebut. Tahap ini menjelaskan pengkodean sistem yang akan dibuat dengan bahasa pemrograman tertentu. Bahasa pemrograman yang dipakai untuk mengembangkan sistem pemesanan dan penjualan kantin filkom menggunakan bahasa pemrograman java mobile android.

#### 3.5.2 Pengujian Sistem

Tahap selanjutnya setelah dilakukan implementasi sistem adalah pengujian sistem. Pengujian sistem dilakukan untuk memastikan apakah sistem yang dijalankan telah memenuhi kebutuhan pengguna atau tidak. Pengujian sistem dilakukan dengan cara *blackbox* dan *compatibility*. Pengujian *blackbox* dilakukan untuk menguji kebutuhan fungsional sistem apakah dapat berjalan dengan baik atau tidak. Pengujian *compatibility* adalah pengujian yang digunakan untuk memeriksa apakah perangkat lunak yang dikembangkan mampu berjalan pada hardware, sistem operasi, aplikasi, ataupun lingkungan jaringan yang berbeda. Pengujian kompatibilitas berfungsi untuk menentukan set lingkungan yang diharapkan dapat menjalankan aplikasi yang dikembangkan. Semakin aplikasi



dapat berjalan di banyak jenis perangkat yang berbeda, maka semakin baik aspek kompatibilitasnya.

### 3.6 Fase *Transition*

Fase *Transition* dilakukan setelah fase *construction* dan merupakan fase paling akhir dalam RUP. Fase ini membahas lebih banyak dalam menginstalasi sebuah sistem yang sudah jadi dan pengujian. Instalasi sistem informasi merupakan pengenalan dan penerapan sistem informasi yang telah jadi pada pemangku kepentingan. Pada sistem informasi pemesanan dan penjualan akan diterapkan pada kantin fakultas ilmu komputer (Filkom) Universitas Brawijaya. Lalu untuk pemangku kepentingannya adalah para mahasiswa filkom sebagai pembeli dan para stan penjual makanan dan minuman pada kantin filkom. Fase *Transition* memiliki tahapan yaitu *user acceptance testing (UAT)* dan Pengujian Waktu. Berikut merupakan penjelasan tahapan dalam fase *Transition* diantaranya:

#### 3.6.1 User Acceptance Testing

*User Acceptance Test (UAT)* adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa software yang telah dikembangkan telah dapat diterima oleh pengguna atau tidak. Dokumen *UAT* berupa kuisisioner yang telah dirancang sesuai test case. Langkah-langkah yang diambil untuk UAT biasanya melibatkan satu atau lebih dari berikut ini:

1. User Acceptance Test (UAT) Perencanaan
2. Merancang UA Uji Kasus
3. Memilih Tim yang akan melaksanakan (UAT) Test Cases
4. Pelaksana Uji Kasus
5. Pengadministrasian yang Cacat ditemukan selama UAT
6. Menyelesaikan masalah / Bug Fixing
7. Sign Off

#### 3.6.2 Pengujian Waktu

Tahap Pengujian Waktu merupakan tahap yang melakukan pengujian dan analisis waktu yang dibutuhkan dalam proses bisnis *as-is* dan proses bisnis *to-be*. Tujuan dari Pengujian Waktu ialah membandingkan dan memastikan alur proses bisnis *to-be* lebih baik/lebih cepat dibanding proses bisnis *as-is*. Perbandingan ini dilakukan pada 3 responden pembeli dengan menggunakan sistem yang baru, lalu dari hasil pengujian tersebut dilakukan perbandingan dan analisis dengan hasil waktu yang dibutuhkan pada proses bisnis *as-is*.

### 3.7 Kesimpulan

Penarikan suatu kesimpulan dilakukan pada tahap akhir penelitian ini. Kesimpulan berisi jawaban dari masalah yang telah dirumuskan. Selain itu pada akhir penulisan terdapat saran yang dapat dijadikan pertimbangan untuk dilakukan pada penelitian selanjutnya.



## BAB 4 ANALISIS KEBUTUHAN

Bab ini membahas pemodelan sistem informasi pemesanan dan penjualan yang akan dikembangkan pada kantin filkom, mulai dari pemodelan proses bisnis (as-is) dan pemodelan sistem (to-be). Dalam bab ini akan membahas Fase *Inseption* pada RUP dan memulai Fase *Elaboration* pada tahap *requirement*. Pada bab ini ada dua tahap, yaitu:

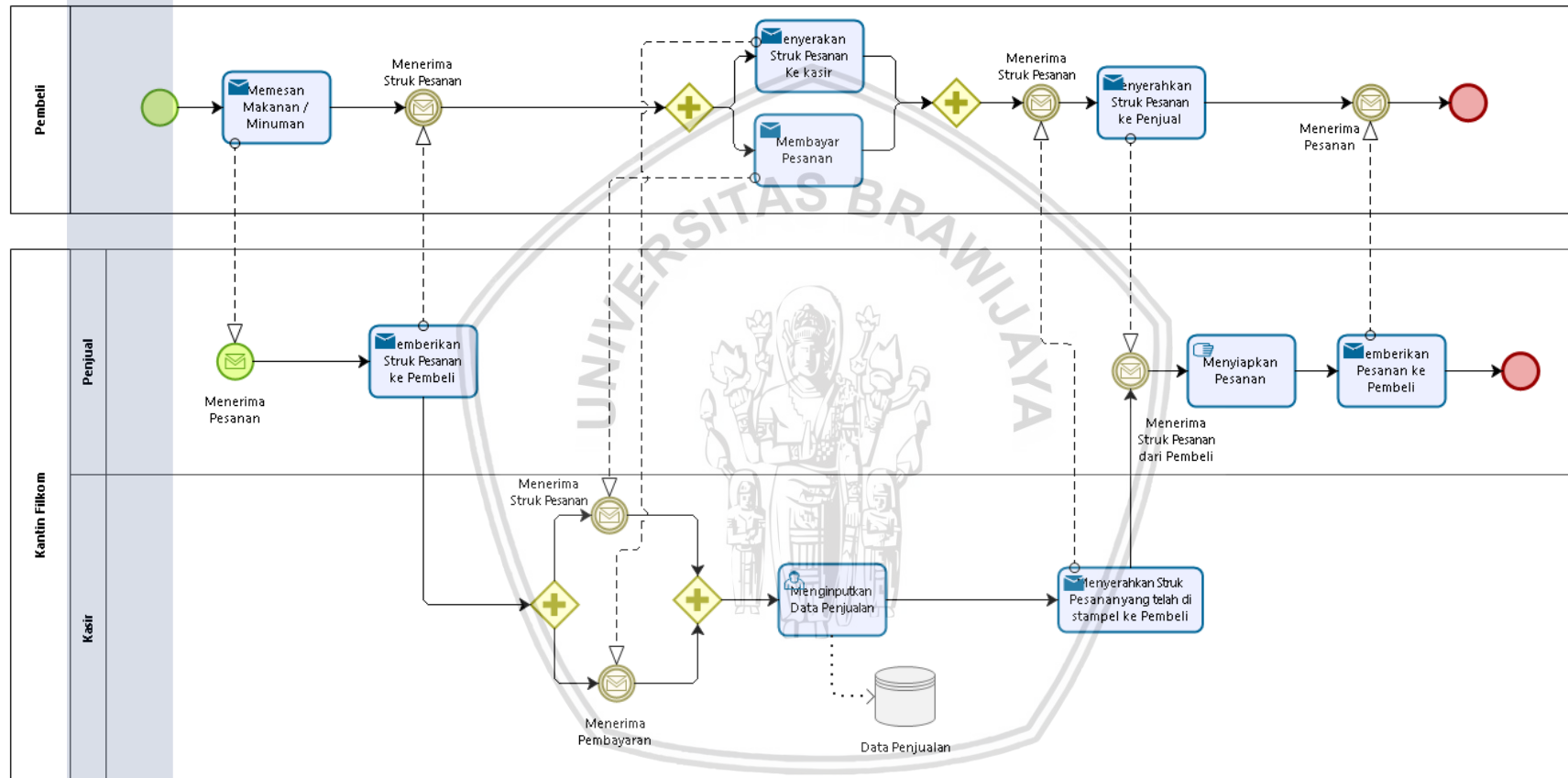
### 4.1 Business Modelling

Business Modelling sebagai suatu teknik pemodelan yang digunakan untuk menggambarkan model sebuah bisnis. Business Modelling digunakan untuk meninjau, meningkatkan dan membuat sebuah bisnis. Model yang digunakan dalam Business Modelling yaitu Business Process Model and Notation. Lalu akan dilakukan Analisa Proses Bisnis dengan cara menganalisa Proses Bisnis yang sedang berjalan saat ini (As-Is) untuk mencari permasalahannya, lalu dari permasalahan tersebut akan didapatkan solusinya dengan menggambarkan Proses Bisnis Usulan (To-Be).

#### 4.1.1 Pemodelan Proses Bisnis As-Is

Pemodelan Proses Bisnis as-is ini untuk menggambarkan Proses Bisnis pemesanan dan penjualan pada kantin Filkom yang sedang berjalan saat ini. Dengan memodelkan proses bisnis as-is penulis dapat melakukan analisis permasalahannya. Dari hasil wawancara terhadap Pengelola Kantin Filkom terdapat pihak-pihak yang terkait yaitu Pembeli, Penjual, dan Kasir lalu didapatkan juga alur proses bisnis seperti berikut:

Pertama pembeli mengantri untuk melakukan pesanan lalu setelah antrian selesai pembeli memesan makanan atau minuman, setelah itu penjual akan mencatat menu pesanan yang dipilih di kertas kecil(struk) lalu memberikannya ke pembeli untuk melakukan pembayaran di kasir, setelah itu kasir akan melakukan pengimputan data penjualan dan memberikan stempel lunas pada kertas pesanan(struk), lalu pembeli kembali lagi ke penjual untuk memberikan struk yang telah diberi stempel lunas kepada penjual, setelah itu penjual menyiapkan makanan atau minuman, setelah makanan telah siap penjual akan memberikan ke pembeli. Alur proses bisnis tersebut seperti yang ditunjukkan pada Gambar 4.1 berikut ini:



Gambar 4.1 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (as-is)

Berdasarkan hasil wawancara dan observasi didapatkan permasalahan pemesanan saat ini dan waktu yang diperlukan untuk melakukan proses pemesanan yang dijelaskan pada Tabel 4.1.

**Tabel 4.1 identifikasi permasalahan dan waktu**

No.	Aktivitas	Waktu	Permasalahan
1	Memesan makanan atau minuman	3 menit	penjual akan kebingungan dalam melayani pembeli jika ramai pembeli
2	Menyerahkan struk pesanan dari penjual ke pembeli sampai Menyerahkan struk pesanan dari pembeli ke penjual lagi	2 menit	Proses pemesanan dan pembayaran yang masih manual
3	Memberikan pesanan ke pelanggan	2 menit	Jika pembeli menunggu digazebo penjual harus mencari pembeli di lokasi gazebo sambil meneriakkan nama pesanan
4	Menunggu pesanan diterima ( <i>intermediate event message</i> menerima pesanan)	6 menit	Saat pembeli menunggu pesanan di depan stan penjual, pembeli harus menunggu antrian pesanan sambil berdiri

Dari table tersebut dapat disimpulkan bahwa proses bisnis pemesanan pada kantin filkom saat ini akan membutuhkan waktu sekiranya 13 menit dalam melakukan pemesanan makanan dan minuman. Oleh sebab itu penulis menyarankan menggunakan proses bisnis usulan. Sebelum memodelkan proses bisnis usulan (*to-be*), perlu menentukan solusi permasalahan tersebut. Solusi permasalahan dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Analisa Domain Masalah dan Solusinya**

No.	Permasalahan	Modul Operational CRM	Solusi
1	penjual akan kebingungan dalam melayani pembeli jika ramai pembeli	<i>Sales force automation (SFA)</i>	Membangun sistem yang dapat melihat siapa saja yang melakukan pemesanan makanan atau minuman
2	Proses pemesanan dan pembayaran yang masih manual	<i>Service Automation (SA)</i>	Membangun sistem yang dapat melayani pembeli dalam memesan dan membayar makanan atau minuman

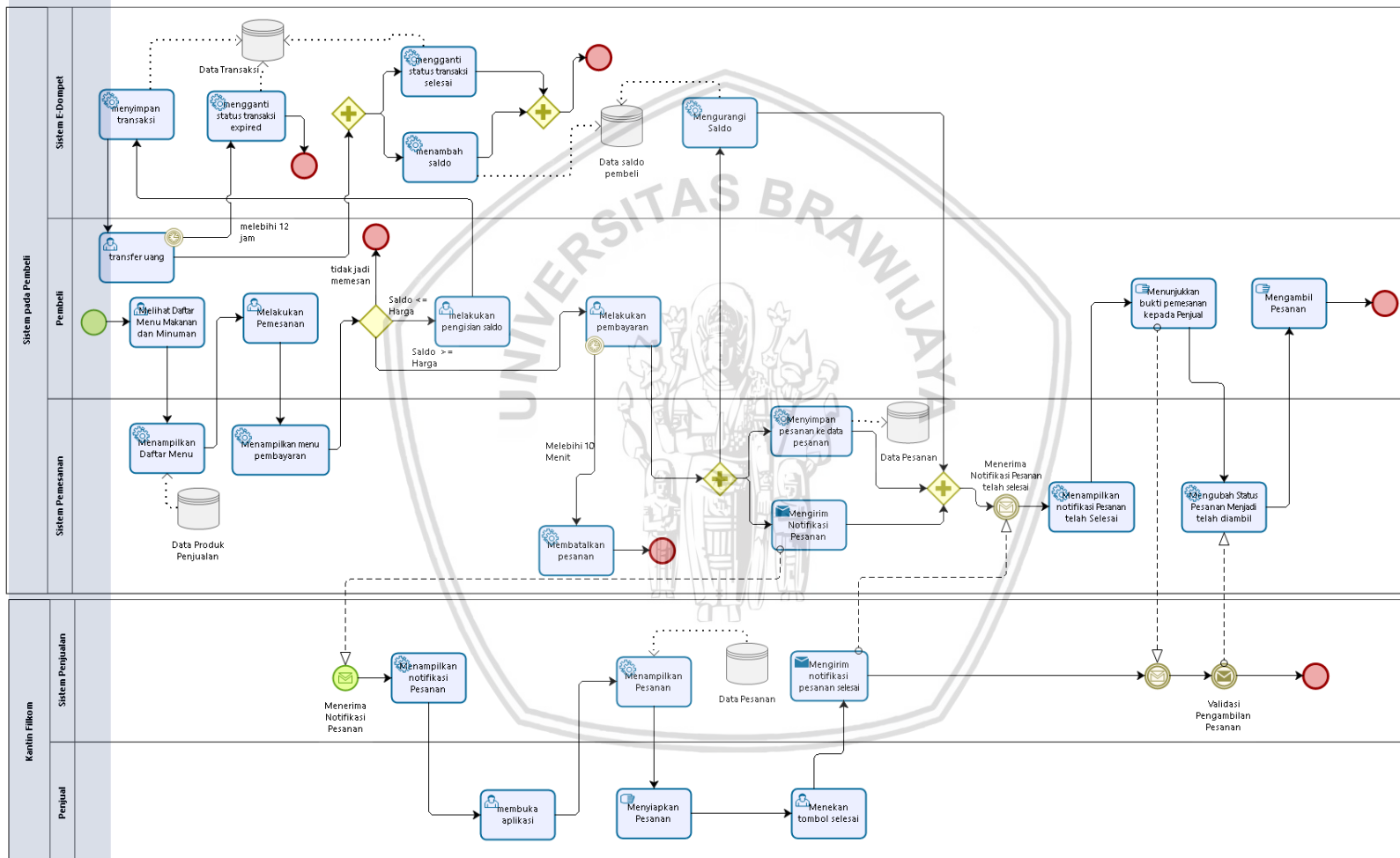
No.	Permasalahan	Modul Operational CRM	Solusi
3	Jika pembeli menunggu digazebo penjual harus mencari pembeli di lokasi gazebo sambil meneriakkan nama pesanan	<i>Sales force automation (SFA)</i>	Membangun sistem yang dapat memberikan informasi jika pesanan telah selesai kepada pembeli supaya pembeli dapat mengambilnya sendiri di stan penjual
4	Saat pembeli menunggu pesanan di depan stan penjual, pembeli harus menunggu antrian pesanan sambil berdiri	<i>Service Automation (SA)</i>	Membangun sistem yang dapat membuat pembeli tidak perlu menunggu didepan stan penjual

Tabel 4.2 menjelaskan permasalahan yang didapatkan setelah melakukan analisis proses bisnis *as-is*. Terdapat 4 permasalahan yang diberikan solusinya menggunakan modul pada operasional CRM yaitu SFA dan SA. Permasalahan pertama menggunakan SFA untuk solusinya, permasalahan kedua menggunakan modul SA, permasalahan ketiga menggunakan modul SFA dan permasalahan terakhir menggunakan SA untuk menangani permasalahan tersebut. Setelah melakukan analisis permasalahan dan memberikan solusi berdasarkan modul operasional CRM selanjutnya melakukan pemodelan proses bisnis rekomendasi (*To-be*).

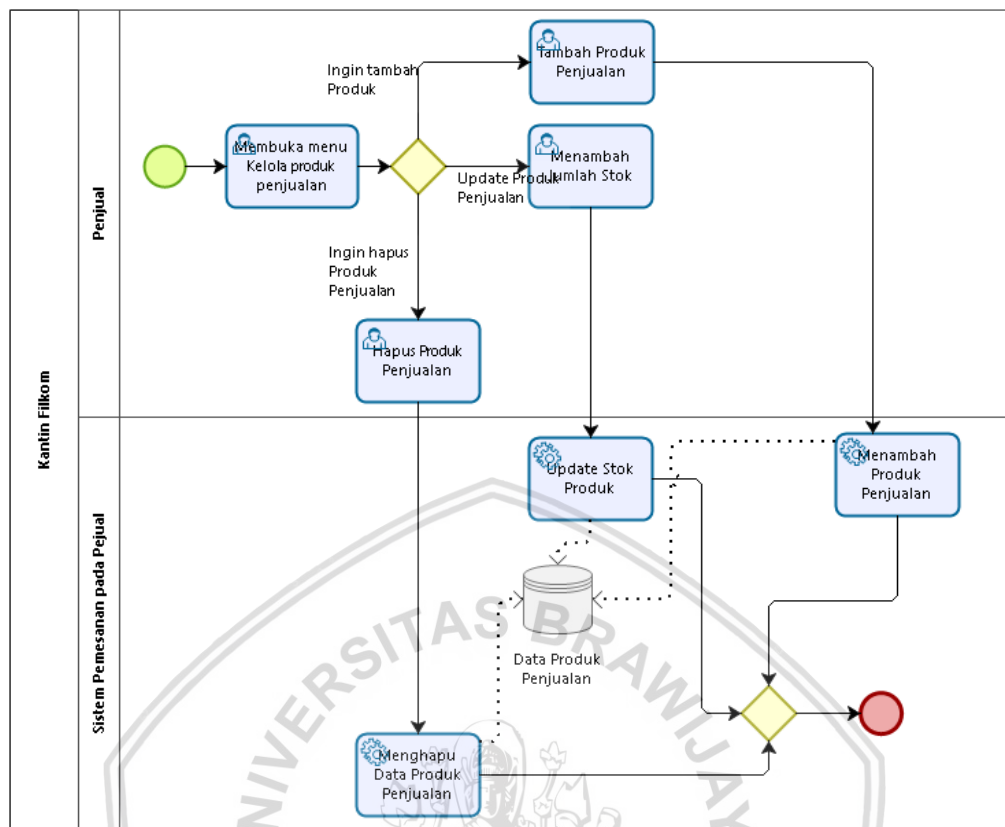
#### 4.1.2 Pemodelan Proses Bisnis To-be

Pemodelan proses bisnis *to-be* digunakan untuk memodelkan proses bisnis usulan yang dapat mengatasi permasalahan yang telah diidentifikasi pada Tabel 4.1. Pada Tabel 4.2 digunakan untuk dasar acuan dalam memodelkan proses bisnis *to-be*. Alur proses bisnis *to-be* pemesanan dan penjualan pada kantin filkom dijelaskan seperti berikut:

Peroses awal ketika pembeli ingin memesan makanan, pembeli terlebih dahulu memilih menu yang tersedia dan melakukan pembayaran pada aplikasi, setelah itu pesanan akan tampil di aplikasi penjual dan penjual akan menyiapkan makanan/minuman yang dipesan, setelah pesanan telah selesai, penjual akan menekan tombol selesai pada aplikasi dan secara otomatis akan mengirim notifikasi pada pembeli jika pesanan telah siap diambil, lalu pembeli akan mengambil pesanan dengan menunjukkan bukti pesanan, setelah itu penjual akan melakukan validasi pengambilan pesanan. Pemodelan proses bisnis usulan pemesanan dan penjualan pada kantin filkom dapat dilihat pada Gambar 4.2. Lalu terdapat proses bisnis dalam mempersiapkan penjualan yang dapat dilihat pada Gambar 4.3.



Gambar 4.2 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (to-be)

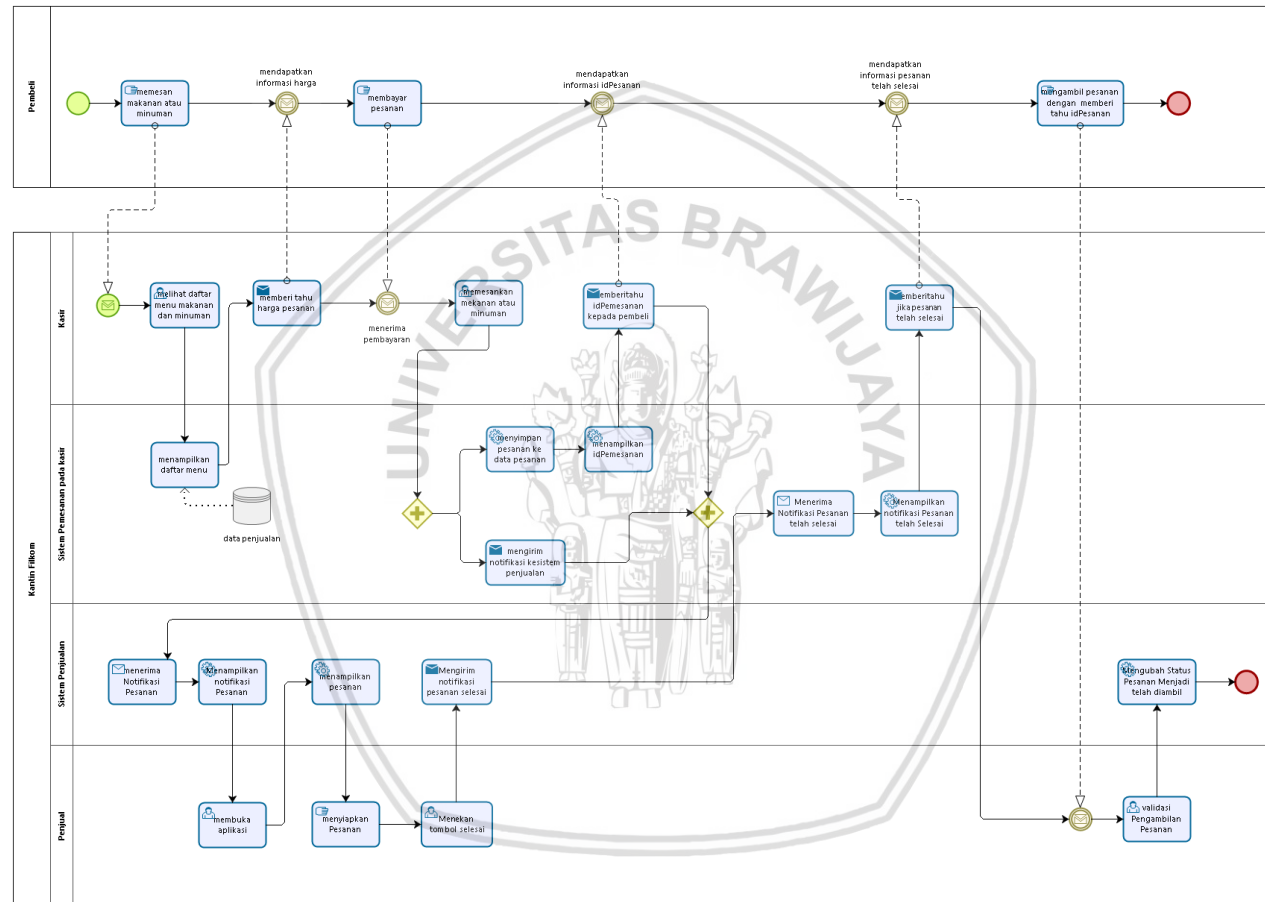


**Gambar 4.3 Bisnis Proses to-be Menyiapkan Penjualan**

Namun untuk masa transisi sistem informasi pemesanan dan penjualan diatas jika ada pembeli yang masih belum memiliki sistem atau aplikasi pemesanan, pembeli dapat melakukannya secara manual. Pembeli hanya perlu memesan pada kasir yang memiliki sistem pemesanannya sendiri yang terintegrasi pada sistem penjualan. Alur proses bisnis to-be pemesanan dan penjualan pada kantin filkom yang dalam masa transisi dijelaskan seperti berikut:

Ketika pembeli ingin memesan makanan atau minuman namun pembeli belum memiliki aplikasi pemesanan makanan dan minuman maka pembeli harus pergi ke kasir untuk melakukan pemesanan dan pembayaran secara manual, setelah itu kasir akan melakukan pemesanan melalui sistem pemesanan khusus kasir, setelah pemesanan telah dilakukan maka kasir akan memberi tahu idpesanan kepada pembeli untuk validasi pengambilan pesanan pada penjual, setelah penjual selesai menyiapkan pesanan, penjual akan menekan tombol selesai pada aplikasi dan secara otomatis akan mengirim notifikasi pada sistem pemesanan pada kasir dan kasir akan memberi tahu kepada pembeli jika pesanan telah selesai, setelah pembeli mengambil pesanan pada penjual sambil memberikan idpesanan pada penjual.





Gambar 4.4 Bisnis Proses Pemesanan dan Penjualan pada Kantin Filkom (to-be) selama masa transisi

### 4.1.3 Analisa Kebutuhan Pemangku Kepentingan

Analisa kebutuhan pemangku kepentingan dilakukan untuk mengetahui apa yang diinginkan oleh pemangku kepentingan untuk mendukung proses bisnis. Analisa kebutuhan pemangku kepentingan bertujuan untuk menemukan solusi untuk sistem yang dapat mendukung target bisnis. Dalam melakukan Analisa kebutuhan pemangku kepentingan terdapat langkah-langkah sebagai berikut:

#### 4.1.3.1 Tipe Pemangku Kepentingan

Tipe pemangku kepentingan adalah kelompok pemangku kepentingan yang memiliki karakteristik dan hubungan dengan sistem atau proyek yang menghasilkan sistem. Tipe pemangku kepentingan pada sistem informasi penjualan dan pemesanan pada kantin filkom dapat dilihat pada Tabel 4.3.

**Tabel 4.3 Tipe Pemangku Kepentingan**

No.	Tipe Pemangku Kepentingan	Deskripsi Pemangku Kepentingan
1	Penjual	seseorang yang menjual atau melayani pemesanan makanan dan minuman kepada pembeli di kantin filkom.
2	Pembeli	sesorang yang melakukan pemesanan makanan atau minuman pada penjual kantin filkom
3	Bank	Pihak yang berhubungan dengan sistem pembayaran yaitu E-Dompet yang bertujuan untuk mengisi saldo pada sistem E-Dompet

#### 4.1.3.2 Peran dan Perwakilan Pemangku Kepentingan

Peran dan perwakilan pemangku kepentingan adalah kelompok pemangku kepentingan yang memiliki peran dan tanggung jawab yang sama pada kegiatan yang dilakukan. Dengan mengetahui peran, maka dapat diambil perwakilan dari pemangku kepentingan. Peran dan perwakilan pemangku kepentingan pada sistem informasi penjualan dan pemesanan pada kantin filkom dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Peran dan Perwakilan Pemangku Kepentingan**

No.	Peran dan Perwakilan Pemangku Kepentingan	Deskripsi
1	Penjual	Orang yang melayani pemesanan makanan dan minuman kepada pembeli di kantin filkom.

No.	Peran dan Perwakilan Pemangku Kepentingan	Deskripsi
2	Pembeli	Orang yang dapat melakukan pemesanan makanan atau minuman pada penjual kantin filkom.

#### 4.1.3.3 Peran Pengguna

Peran pengguna merupakan peran yang dijalankan pengguna saat melakukan interaksi dengan sistem. Peran pengguna pada sistem informasi penjualan dan pemesanan pada kantin filkom dapat dilihat pada Tabel 4.5.

**Tabel 4.5 Peran Pengguna**

No.	Peran Pengguna	Deskripsi Pengguna
1	Penjual	Penjual merupakan pengguna yang dapat melayani pemesanan makanan atau minuman di kantin filkom dari pagi sampai sore
2	Pembeli	Pembeli merupakan pengguna yang dapat melakukan pemesanan makanan atau minuman pada penjual kantin filkom.
3	User	User merupakan pengguna sistem yang dapat melakukan login.

## 4.2 Requirement

Requirement berfungsi untuk menggambarkan sistem apa yang harus dilakukan. Tahap ini menandakan bahwa telah dimulai Fase *Elaboration*. Dari hasil business modelling dapat dilakukan identifikasi aktor dan dari aktor tersebut dapat digali kebutuhan sistemnya.

### 4.2.1 Identifikasi Aktor

**Tabel 4.6 indentifikasi Aktor**

No.	Kode Aktor	Nama Aktor	Deskripsi Aktor
1	AK-01	Penjual	Penjual adalah seseorang yang menjual makanan atau minuman di kantin filkom dari pagi sampai sore
2	AK-02	Pembeli	Pembeli adalah orang yang melakukan pemesanan makanan pada stan penjual

No.	Kode Aktor	Nama Aktor	Deskripsi Aktor
			kantin filkom
3	AK-03	User	User merupakan pengguna sistem yang dapat melakukan login.
4	AK-04	Bank	Pihak yang berhubungan dengan sistem pembayaran yaitu E-Dompet yang bertujuan untuk mengisi saldo pada sistem E-Dompet

#### 4.2.2 User Requirement

Tabel 4.7 Kebutuhan Pengguna

No.	[AK-01] Penjual	
	Kode UR	Requirement
1	UR-01	Mengetahui pesanan dari pembeli secara <i>realtime</i>
2	UR-02	Memberi tahu kepada pembeli jika pesanan telah selesai
3	UR-03	Mengelola produk penjualan
4	UR-07	Mengetahui hasil penjualan
No.	[AK-02] Pembeli	
	Kode UR	Requirement
1	UR-04	Memesan makakanan atau minuman
2	UR-05	Melakukan pembayaran
3	UR-08	Mengelola saldo
4	UR-09	Mendapatkan Pesanannya

No.	[AK-03] User	
	Kode UR	Requirement
1	UR-06	Dapat masuk kedalam sistem

#### 4.2.3 Fitur Sistem Informasi

**Tabel 4.8 Fitur Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom**

No	Kode Fitur	Deskripsi
1	FSI-01	Sistem dapat menyediakan layanan yang dapat melayani pembeli dalam memesan dan membayar makanan atau minuman secara terotomatisasi.
2	FSI-02	Sistem menyediakan layanan yang dapat memberikan informasi jika pesanan telah selesai kepada pembeli supaya pembeli dapat mengambilnya sendiri di stan penjual.
3	FSI-03	Sistem menyediakan layanan yang dapat mengatur pemesanan makanan atau minuman dari pembeli dan produk yang akan dijual.
4	FSI-04	Sistem menyediakan layanan untuk perakapan hasil penjualan yang telah selesai dilakukan dan dapat dilihat dalam periodik hari.
5	FSI-05	Sistem menyediakan layanan untuk dapat masuk kedalam sistem penjualan dan pembelian dengan akun setiap pengguna yang berbeda.

#### 4.2.4 Software Requirement Specification

##### Functional Requirement

**Tabel 4.9 SRS Functional**

[AK01] Penjual				
No.	Kode UR	Kode Fitur	Kode SRS	Software Requirement Statement
1	UR-01	FSI-03	SRF-01	Melihat pesanan

2	UR-02	FSI-02	SRF-02	Edit status pesanan selesai
3	UR-03	FSI-03	SRF-03	Tambah produk
4	UR-03	FSI-03	SRF-04	Update produk
5	UR-03	FSI-03	SRF-05	Hapus produk
6	UR-07	FSI-04	SRF-11	Melihat hasil Penjualan (tambahan pada iterasi ke-1 fase elaborasi)
7	UR-03	FSI-03	SRF-14	Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi)
8	UR-02	FSI-02	SRF-15	Edit status pesanan dibatalkan (tambahan pada iterasi ke-1 fase konstruksi)
[AK-02] Pembeli				
9	UR-04	FSI-01	SRF-06	Melihat menu produk penjualan
10	UR-04	FSI-01	SRF-07	Memesan produk penjualan
11	UR-05	FSI-01	SRF-08	Membayar pesanan
12	UR-08	FSI-01	SRF-09	Mengisi saldo
13	UR-08	FSI-01	SRF-12	Melihat profil akun (tambahan pada iterasi ke-1 fase elaborasi)
14	UR-09	FSI-02	SRF-16	Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi)
[AK-03] User				
15	UR-06	FSI-05	SRF-10	Login
16	UR-06	FSI-05	SRF-13	Registrasi



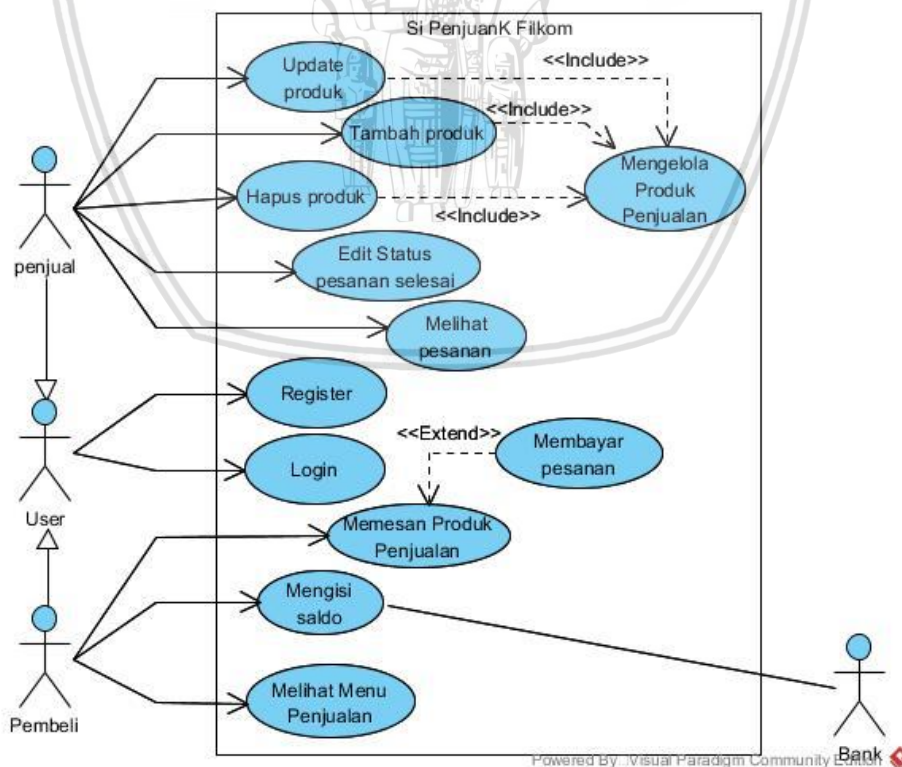
## Non-Functional Requirement

**Tabel 4.10 SRS Non-Functional**

No.	Kode UR	Kode Fitur	Kode SRS	Software Requirement Statement
1	UR-06	FSI-05	SRN-01	Security untuk pengamanan login dan biodata yang tersimpan pada server database
2	UR-01	FSI-01, FSI-02	SRN-02	Menampilkan data dari <i>database</i> server secara <i>realtime</i>

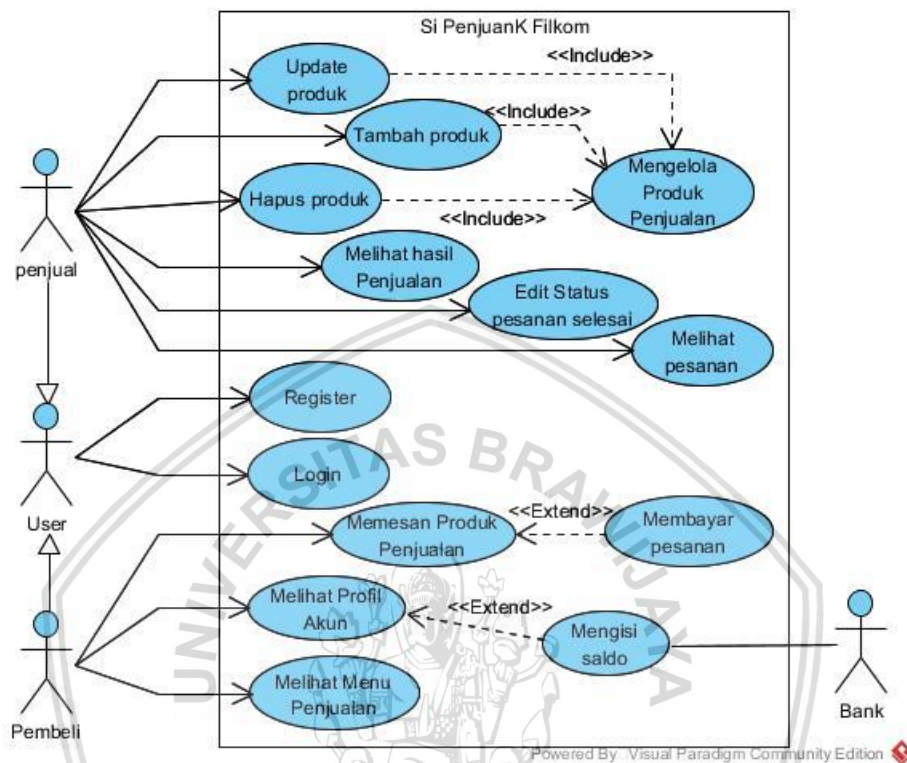
### 4.2.5 Use Case Diagram

Use-case diagram dapat digambarkan Setelah melakukan analisis fungsional pada sistem informasi pemesanan dan penjualan pada kantin filkom. Use case pada Use-case diagram merupakan fungsional sistem tersebut dan aktor Use-case diagram merupakan pengguna ataupun sistem eksternal yang menggunakan sistem tersebut. Berikut Use-case diagram pada sistem informasi pemesanan dan penjualan pada kantin filkom yang dapat dilihat pada Gambar 4.5.



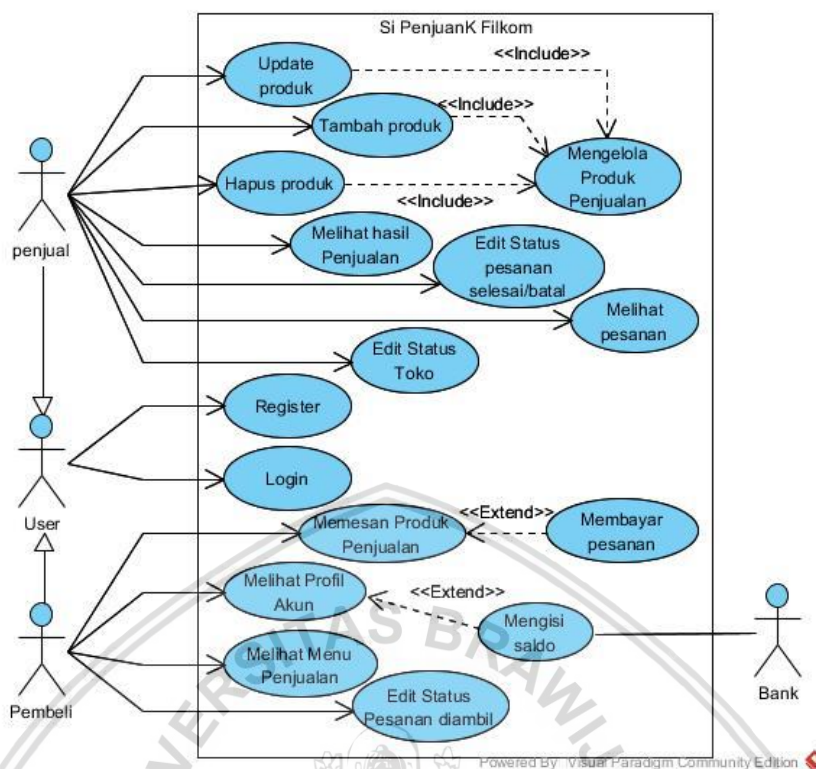
**Gambar 4.5 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom**

Setelah melakukan Analisa ulang dengan pengelola kantin filkom, terdapat penambahan fungsional-fungsional yang diperlukan. Fungsional tersebut adalah Melihat Hasil Penjualan dan Melihat Profil Akun. Berikut Use-case diagram yang telah ditambahkan pada Gambar 4.6.



**Gambar 4.6 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Iterasi ke-1 Fase Elaborasi)**

Setelah masuk fase konstruksi dan melakukan konsultasi kepada para pemangku kepentingan ternyata didapatkan fungsional baru yang diperlukan. Fungsional tersebut adalah Edit Status Toko, Edit Status Pesanan dibatalkan dan Edit Status Pesanan diambil. Fungsional-fungsional tersebut ditambahkan pada iterasi ke-1 fase konstruksi. Berikut Use-case diagram yang telah ditambahkan pada Gambar 4.7.



**Gambar 4.7 Usecase Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Iterasi ke-1 Fase Konstruksi)**

#### 4.2.6 Use Case Scenario

Hasil use case diagram diatas dapat dipetakan menjadi skenario use case pada setiap use case yang ada. Skenario use case merupakan penjabaran alur kerja atau langkah-langkah tiap use case. Skenario dari use case diagram dapat dilihat pada Tabel 4.11 sampai Tabel 4.25.

**Tabel 4.11 Use case scenario Melihat Pesanan**

<i>Use Case Name</i>	Melihat Pesanan
<i>Actor:</i>	Penjual
<i>Brief Description:</i>	Use case ini untuk penjual agar dapat melihat siapa saja pembeli yang memesan produknya
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual membuka aplikasi</li> <li>2. Sistem menampilkan daftar pesanan pembeli</li> </ol>
<i>Alternative Flow - 1:</i>	Pada main flow poin 1 jika tidak ada pesanan

	1. Sistem tidak menampilkan daftar pesanan pembeli
<i>Postconditions:</i>	-

Tabel 4.11 menjelaskan skenario aktor dalam melihat pesanan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika penjual telah melakukan *login* pada sistem. Pada *main flow* ketika penjual membuka aplikasi, sistem akan menampilkan daftar pesanan. Terdapat alternatif kondisi ketika pada *main flow* poin 1 jika tidak ada pesanan maka sistem tidak akan menampilkan daftar pesanan.

**Tabel 4.12 Use case scenario Edit Status pesanan selesai**

<i>Use Case Name</i>	Edit Status pesanan selesai
<i>Actor:</i>	Penjual
<i>Brief Description:</i>	use case ini digunakan untuk memberi tahu pembeli jika pesanan telah selesai dan dapat diambil
<i>Preconditions:</i>	Status dari pesanan yang belum selesai
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual klik tombol selesai</li> <li>2. Sistem mengubah Status menjadi selesai</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Status Pesanan berubah menjadi Selesai

Tabel 4.12 menjelaskan skenario aktor dalam melakukan edit status pesanan menjadi selesai. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika terdapat pesanan dengan status proses/belum selesai. Pada *main flow* ketika penjual menekan tombol selesai, sistem akan mengubah Status menjadi selesai. Hasil akhir dari skenario ini adalah status pesanan yang telah berubah menjadi selesai.

**Tabel 4.13 Use case scenario Melihat Menu Penjualan**

<i>Use Case Name</i>	Melihat Menu Penjualan
<i>Actor:</i>	Pembeli
<i>Brief Description:</i>	Use case ini digunakan agar pembeli dapat melihat daftar menu yang tersedia dari tiap penjual
<i>Preconditions:</i>	Telah user terautentikasi sebagai pembeli
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli membuka aplikasi</li> </ol>

	2. Sistem menampilkan menu penjualan berdasarkan Penjualnya
<i>Alternative Flow - 1:</i>	Pada main flow poin 1 jika tidak ada produk yang dijual 1. Sistem tidak menampilkan produk yang dijual
<i>Postconditions:</i>	-

Tabel 4.13 menjelaskan skenario aktor dalam melihat menu penjualan. Aktor yang berperan pada skenario ini adalah Pembeli. Pembeli dapat melakukan skenario ini ketika pembeli telah melakukan *login* pada sistem. Pada *main flow* ketika pembeli membuka aplikasi, sistem akan menampilkan menu penjualan berdasarkan Penjualnya. Terdapat alternatif kondisi ketika pada *main flow* poin 1 jika tidak ada produk yang dijual maka sistem tidak akan menampilkan produk yang dijual.

**Tabel 4.14 Use case scenario Memesan Produk Penjualan**

<i>Use Case Name</i>	Memesan Produk Penjualan
<i>Actor:</i>	Pembeli
<i>Brief Description:</i>	Use case ini digunakan agar pembeli dapat memesan produk yang dijual oleh penjual
<i>Preconditions:</i>	Terdapat List Penjualan/Stok produk tidak kosong
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli memilih produk dan jumlah pesanan lalu menekan tombol pesan</li> <li>2. Sistem menyimpan data pesanan ke daftar produk yang dipesan</li> <li>3. Sistem menampilkan menu pembayaran</li> </ol>
<i>Alternative Flow - 1:</i>	Pada main flow poin 1 jika produk telah habis maka: <ol style="list-style-type: none"> <li>1. Pembeli mencari produk lainnya</li> <li>2. Kembalik ke main flow poin 1</li> </ol>
<i>Alternative Flow – 2:</i>	Pada main flow poin 2 jika ingin memesan produk lain <ol style="list-style-type: none"> <li>1. Kembali ke main flow poin 1</li> </ol>
<i>Postconditions:</i>	Pesanan pada keranjang bertambah



Tabel 4.14 menjelaskan skenario aktor dalam memesan produk penjualan. Aktor yang berperan pada skenario ini adalah Pembeli. Pembeli dapat melakukan skenario ini ketika terdapat list penjualan yang tersedia atau stok produk tidak kosong. Pada *main flow* ketika Pembeli memilih produk dan jumlah pesanan lalu menekan tombol pesan, sistem akan menyimpan data pesanan ke daftar produk yang dipesan dan menampilkan menu pembayaran. Terdapat alternatif kondisi ketika pada *main flow* poin 1 jika produk telah habis maka Pembeli mencari produk lainnya dan kembali pada *main flow* poin 1, lalu ketika pada *main flow* poin 2 jika ingin memesan produk lain maka kembali ke *main flow* poin 1. Hasil akhir dari skenario ini adalah pesanan yang dipesan bertambah dan masuk di keranjang.

**Tabel 4.15 Use case scenario Membayar Pesanan**

<i>Use Case Name</i>	Membayar Pesanan
<i>Actor:</i>	Pembeli
<i>Brief Description:</i>	Use case ini digunakan untuk mengubah status pesanan yang sebelumnya sedang proses menjadi selesai
<i>Preconditions:</i>	Status pesanan yang belum dibayar, Sudah memesan
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli menekan menu pesanan</li> <li>2. Sistem menampilkan daftar produk yang dipesan</li> <li>3. Pembeli menekan tombol bayar</li> <li>4. Sistem menampilkan menu pembayaran</li> <li>5. Pembeli melakukan pembayaran</li> <li>6. Sistem mengubah status pesanan menjadi telah dibayar</li> <li>7. Sistem memotong saldo pembeli</li> </ol>
<i>Alternative Flow - 1:</i>	<p>Pada main flow poin 1 jika saldo yang dimiliki kurang maka:</p> <ol style="list-style-type: none"> <li>1. Pembeli melakukan flow pada use case Mengisi Saldo E-Dompet</li> <li>2. Kembali ke main flow poin 1</li> </ol>
<i>Postconditions:</i>	<ol style="list-style-type: none"> <li>1. Status pesanan telah dibayar</li> <li>2. Saldo berkurang</li> </ol>

Tabel 4.15 menjelaskan skenario aktor dalam melakukan pembayaran pesanan. Aktor yang berperan pada skenario ini adalah Pembeli. Pembeli dapat



melakukan skenario ini ketika terdapat pesanan yang dipesan/sudah memesan dan status pesanan masih belum dibayar. Pada *main flow* ketika Pembeli menekan menu pesanan, sistem akan menampilkan daftar produk yang dipesan, lalu ketika pembeli menekan tombol bayar, sistem akan menampilkan menu pembayaran, selanjutnya ketika pembeli melakukan pembayaran, sistem akan mengubah status pesanan menjadi telah dibayar dan memotong saldo pembeli. Terdapat alternatif kondisi ketika pada *main flow* poin 1 jika saldo yang dimiliki kurang maka Pembeli melakukan flow pada use case Mengisi Saldo E-Dompet lalu kembali ke *main flow* poin 1. Hasil akhir dari skenario ini adalah status pesanan menjadi telah dibayar dan saldo pembeli berkurang.

**Tabel 4.16 Use case scenario Melihat Profil Akun (tambahan pada iterasi pertama fase elaborasi)**

<i>Use Case Name</i>	Melihat Profil Akun
<i>Actor:</i>	Pembeli
<i>Brief Description:</i>	Use case ini digunakan untuk melihat profil pembeli dan jumlah saldo yang dimiliki
<i>Preconditions:</i>	Telah user terautentikasi sebagai pembeli
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli menekan profil saya</li> <li>2. Sistem menampilkan profil pembeli</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	-

Tabel 4.16 menjelaskan skenario aktor dalam melihat profil akun pengguna. Aktor yang berperan pada skenario ini adalah Pembeli. Pembeli dapat melakukan skenario ini ketika pembeli telah melakukan login. Pada *main flow* ketika Pembeli menekan profil saya, sistem akan menampilkan profil pembeli.

**Tabel 4.17 Use case scenario Mengisi Saldo E-Dompet**

<i>Use Case Name</i>	Mengisi Saldo E-Dompet
<i>Actor:</i>	Pembeli, bank
<i>Brief Description:</i>	Use case ini digunakan untuk mengisi saldo e-dompet agar dapat digunakan untuk memesan produk penjualan kantin
<i>Preconditions:</i>	Telah user terautentikasi sebagai pembeli dan berada di profil saya.
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli menekan tambah saldo</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan form tambah saldo</li> <li>3. Pembeli mengisi form dan menekan konfirm</li> <li>4. Sistem menampilkan status deposit pada riwayat deposit dan menunggu proses transfer</li> <li>5. Pembeli melakukan transfer melalui bank yg dituju</li> <li>6. Sistem menambah saldo dan mengganti status deposit menjadi selesai</li> </ol>
<i>Alternative Flow - 1:</i>	<p>Pada main flow poin 3 jika pembeli mengosongkan salah satu inputan:</p> <ol style="list-style-type: none"> <li>1. Sistem akan memberi peringatan jika pembeli harus mengisi semua inputan.</li> <li>2. Kembali ke main flow poin 3</li> </ol>
<i>Postconditions:</i>	<ol style="list-style-type: none"> <li>1. Saldo bertambah</li> <li>2. Riwayat deposit bertambah</li> </ol>

Tabel 4.17 menjelaskan skenario aktor dalam mengisi saldo E-Dompet. Aktor yang berperan pada skenario ini adalah Pembeli dan bank sebagai perantaranya. Pembeli dapat melakukan skenario ini ketika pembeli telah melakukan login. Pada *main flow* ketika Pembeli menekan tambah saldo, sistem akan menampilkan form tambah saldo, lalu ketika pembeli mengisi form dan menekan konfirm, sistem akan menampilkan status deposit pada riwayat deposit dan menunggu proses transfer, selanjutnya ketika pembeli melakukan transfer melalui bank yg dituju, sistem akan menambah saldo dan mengganti status deposit menjadi selesai. Terdapat alternatif kondisi ketika pada *main flow* poin 3 jika pembeli mengosongkan salah satu inputan maka sistem akan memberi peringatan jika pembeli harus mengisi semua inputan lalu kembali ke *main flow* poin 3. Hasil akhir dari skenario ini adalah saldo bertambah dan Riwayat deposit bertambah.

**Tabel 4.18 Use case scenario Melihat Hasil Penjualan (tambahan pada iterasi ke-1 fase elaborasi)**

<i>Use Case Name</i>	Melihat Hasil Penjualan
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk menampilkan semua hasil penjualan yang telah dilakukan
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual menekan menu Hasil</li> </ol>

	2. Sistem menampilkan hasil Penjualan pada hari ini 3. Penjual menentukan <i>range</i> tanggal 4. Sistem menampilkan hasil Penjualan pada <i>range</i> tanggal tersebut
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	-

Tabel 4.18 menjelaskan skenario aktor dalam melihat hasil penjualan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika penjual telah melakukan login. Pada *main flow* ketika Penjual menekan menu hasil, sistem akan menampilkan hasil penjualan pada hari ini, lalu ketika penjual menentukan *range* tanggal, sistem akan menampilkan hasil Penjualan pada *range* tanggal tersebut.

**Tabel 4.19 Use case scenario Mengelola Produk Penjualan**

<i>Use Case Name</i>	Mengelola Produk Penjualan
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk mengelola produk penjualan penjual
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual
<i>Main Flow:</i>	5. Penjual menekan tombol menu kelola produk 6. Sistem menampilkan opsi kelola produk
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	-

Tabel 4.19 menjelaskan skenario aktor dalam mengelola produk penjualan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika pembeli telah melakukan login. Pada *main flow* ketika Penjual menekan tombol menu kelola produk, sistem akan menampilkan opsi kelola produk.

**Tabel 4.20 Use case scenario Tambah Produk**

<i>Use Case Name</i>	Tambah Produk
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk menambah produk penjualan penjual

<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual menekan tombol tambah produk</li> <li>2. Sistem menampilkan Form tambah produk</li> <li>3. Penjual mengisi Form tambah produk dan menekan tambah</li> <li>4. Sistem menambah produk</li> </ol>
<i>Alternative Flow - 1:</i>	<p>Pada <i>main flow</i> poin 3 jika ada salah satu inputan tidak diisi:</p> <ol style="list-style-type: none"> <li>1. Sistem akan menampilkan peringatan penjual harus mengisi semua inputan.</li> <li>2. Kembali ke <i>main flow</i> poin 3</li> </ol>
<i>Postconditions:</i>	Produk bertambah

Tabel 4.20 menjelaskan skenario aktor dalam melakukan penambahan produk penjualan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika pembeli telah melakukan login. Pada *main flow* ketika Penjual menekan tombol tambah produk, sistem akan menampilkan form tambah produk, lalu ketika penjual mengisi form tambah produk dan menekan tambah, sistem akan menambah produk. Terdapat alternatif kondisi ketika pada *main flow* poin 3 jika penjual mengosongkan salah satu inputan maka sistem akan memberi peringatan jika penjual harus mengisi semua inputan lalu kembali ke *main flow* poin 3. Hasil akhir dari skenario ini adalah produk penjualan bertambah.

**Tabel 4.21 Use case scenario Hapus Produk**

<i>Use Case Name</i>	Hapus Produk
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk menghapus produk penjualan penjual
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual, Terdapat minimal satu produk
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual menekan tombol hapus produk</li> <li>2. Sistem manghapus produk</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Produk berkurang

Tabel 4.21 menjelaskan skenario aktor dalam melakukan hapus produk penjualan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat

melakukan skenario ini ketika pembeli telah melakukan login dan terdapat minimal satu produk penjualan. Pada *main flow* ketika Penjual menekan tombol hapus produk, sistem akan menghapus produk. Hasil akhir dari skenario ini adalah produk penjualan berkurang.

**Tabel 4.22 Use case scenario Update Produk**

<i>Use Case Name</i>	Update Produk
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk mengubah stok produk penjualan penjual
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual, Terdapat minimal satu produk
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual menekan tombol <i>update</i> produk</li> <li>2. Sistem menampilkan Form update produk</li> <li>3. Penjual mengisi Form update produk</li> <li>4. Sistem mengubah produk</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Produk telah diupdate

Tabel 4.22 menjelaskan skenario aktor dalam melakukan update produk penjualan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika pembeli telah melakukan login dan terdapat minimal satu produk penjualan. Pada *main flow* ketika Penjual menekan tombol *update* produk, sistem akan menampilkan form update produk, lalu ketika penjual mengisi form update produk dan menekan ubah, sistem akan mengubah produk. Hasil akhir dari skenario ini adalah informasi produk telah berubah.

**Tabel 4.23 Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi)**

<i>Use Case Name</i>	Edit Status Toko
<i>Actor:</i>	Penjual
<i>Description:</i>	Use case ini digunakan untuk mengubah status toko agar pembeli dapat mengetahui toko sedang buka.
<i>Preconditions:</i>	Telah user terautentikasi sebagai penjual
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual menekan/menggeser tombol saklar/<i>swich</i> buka toko</li> <li>2. Sistem mengubah status toko</li> </ol>

<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Status toko telah diupdate

Tabel 4.23 menjelaskan skenario aktor dalam melakukan edit status toko. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika pembeli telah melakukan login. Pada *main flow* ketika penjual menekan/menggeser tombol saklar/*swich* buka toko, sistem akan mengubah status toko. Hasil akhir dari skenario ini adalah informasi status toko telah berubah.

**Tabel 4.24 Use case scenario Edit status pesanan dibatalkan (tambahan pada iterasi ke-1 fase konstruksi)**

<i>Use Case Name</i>	Edit status pesanan dibatalkan
<i>Actor:</i>	Penjual
<i>Brief Description:</i>	use case ini digunakan untuk memberi tahu pembeli jika pesanan dibatalkan
<i>Preconditions:</i>	Status dari pesanan yang belum selesai/proses
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Penjual klik tombol "x"</li> <li>2. Sistem mengubah Status menjadi dibatalkan dan mengembalikan nominal saldo ke pembeli</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Status Pesanan berubah menjadi Dibatalkan dan nominal saldo pembeli bertambah sesuai harga produk yang dibatalkan.

Tabel 4.24 menjelaskan skenario aktor dalam melakukan edit status pesanan menjadi dibatalkan. Aktor yang berperan pada skenario ini adalah Penjual. Penjual dapat melakukan skenario ini ketika terdapat pesanan dengan status proses/belum selesai. Pada *main flow* ketika penjual menekan tombol "x", sistem akan mengubah Status menjadi dibatalkan dan mengembalikan nominal saldo ke pembeli. Hasil akhir dari skenario ini adalah status pesanan yang telah berubah menjadi dibatalkan dan nominal saldo pembeli bertambah sesuai harga produk yang dibatalkan.

**Tabel 4.25 Use case scenario Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi)**

<i>Use Case Name</i>	Edit Status Pesanan Diambil
<i>Actor:</i>	Pembeli



<i>Description:</i>	Use case ini digunakan untuk mengubah status pesanan menjadi telah diambil agar penjual dapat mengetahui pesanan mana yang akan diambil.
<i>Preconditions:</i>	Telah user terautentikasi sebagai pembeli, status pesanan telah selesai
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pembeli menekan tombol ambil pada menu riwayat</li> <li>2. Sistem menampilkan pesanan yang akan diambil.</li> <li>3. Penjual mengecek dan menekan tombol telah diambil.</li> <li>4. Sistem mengubah status pesanan telah diambil.</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Status Pesanan berubah menjadi telah diambil.

Tabel 4.25 menjelaskan skenario aktor dalam melakukan pengambilan produk pesanan. Aktor yang berperan pada skenario ini adalah Pembeli. Pembeli dapat melakukan skenario ini ketika pembeli telah melakukan login dan status pesanan telah selesai. Pada *main flow* ketika Pembeli menekan tombol ambil pada menu riwayat, sistem akan menampilkan pesanan yang akan diambil, lalu ketika penjual mengecek dan menekan tombol telah diambil, sistem akan mengubah status pesanan telah diambil. Hasil akhir dari skenario ini adalah informasi status pesanan berubah menjadi telah diambil.

## BAB 5 PERANCANGAN SISTEM

Bab ini berisi perancangan dari sistem yang akan dibangun, perancangan sistem ini terdiri atas perancangan *activity diagram*, *sequence diagram*, *class diagram*, *physical data model*, dan perancangan antarmuka. Dalam bab ini akan lebih banyak membahas Fase *Elaboration* pada RUP. Pada bab ini terdapat tahapan *Analysis & Design* dan iterasinya.

Analisis dan desain adalah Teknik menganalisa dan memodelkan suatu sistem. Analisis dan desain bertujuan agar memudahkan programmer dalam melakukan pengembangan sistem. Untuk dapat memodelkan sistem dengan baik maka pemodelan sistem tersebut harus konsisten dengan kebutuhan yang telah didapatkan.

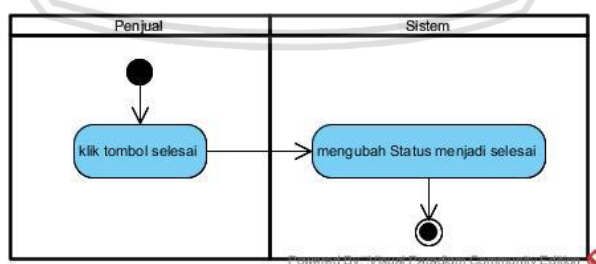
Bab ini dimulai dengan iterasi ke-1 pada fase elaborasi. Iterasi ini adalah iterasi dimana pengembang melakukan tahap selanjutnya setelah fase insepisi dan memulai melakukan tahapan perancangan sistem yang akan dibuat. Iterasi pertama pada fase elaborasi dimulai pada use-case diagram jika terdapat penambahan fungsional sistem.

### 5.1 Activity Diagram

Perancangan activity diagram berisi penggambaran aliran kerja dari sistem. Berikut ini merupakan activity diagram sistem pemesanan dan penjualan kantin filkom:

#### 1. Activity Diagram Edit Status Pesanan Selesai

Ketika penjual selesai menyiapkan pesanan, penjual menekan tombol selesai pada aplikasi, setelah itu sistem akan mengubah status pesanan menjadi selesai. *Activity Diagram* ini digunakan agar pembeli dapat mengetahui pesanan telah selesai atau belum. Berikut ini *Activity Diagram* Edit Status Pesanan Selesai dapat dilihat pada Gambar 5.1.

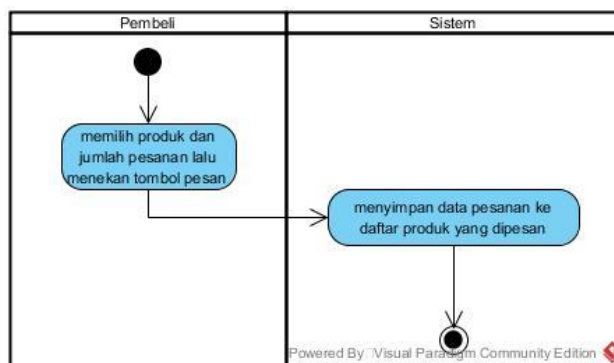


Gambar 5.1 Activity Diagram Edit Status Pesanan Selesai

#### 2. Activity Diagram Memesan Produk Penjualan

Ketika ingin memesan makanan atau minuman pembeli memilih dan menentukan jumlah pesanan lalu menekan tombol pesan, setelah itu sistem akan menyimpan data pesanan ke daftar produk yang dipesan dan menampilkan produk

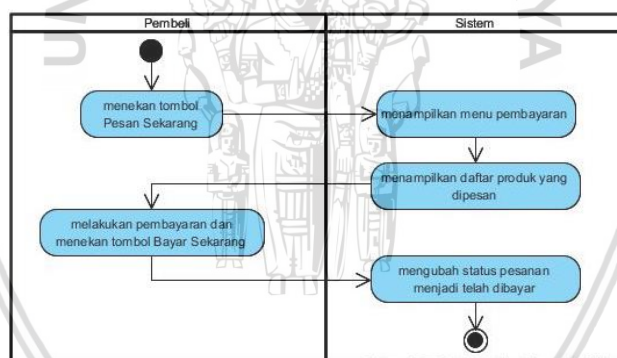
yang dipesan. Berikut ini *Activity Diagram* Memesan Produk Penjualan dapat dilihat pada Gambar 5.2.



**Gambar 5.2 Activity Diagram Memesan Produk Penjualan**

### 3. *Activity Diagram* Membayar Pesanan

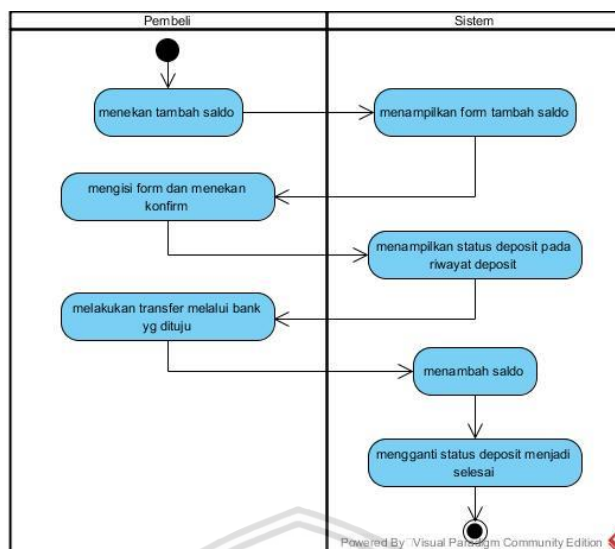
Ketika pembeli ingin melakukan pembayaran maka pembeli menekan menu pesanan. Setelah itu sistem akan menampilkan daftar pesanan. Pembeli menekan tombol bayar. Sistem menampilkan menu pembayaran. Pembeli melakukan pembayaran. Sistem mengubah status pesanan menjadi telah dibayar dan saldo berkurang. Berikut ini *Activity Diagram* Membayar Pesanan dapat dilihat pada Gambar 5.3.



**Gambar 5.3 Activity Diagram Membayar Pesanan**

### 4. *Activity Diagram* Mengisi Saldo E-Dompet

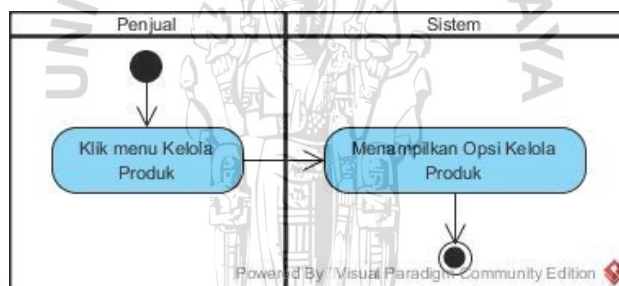
Pada menu profil pembeli menekan tombol “+” untuk melakukan pengisian saldo. Sistem menampilkan form tambah saldo. Pembeli mengisi form dan menekan confirm. Sistem menampilkan status deposit pada riwayat deposit. Pembeli melakukan transfer melalui bank yang dituju. Sistem menambah saldo dan mengganti status deposit menjadi selesai. Berikut ini *Activity Diagram* Mengisi Saldo E-Dompet dapat dilihat pada Gambar 5.4.



**Gambar 5.4 Activity Diagram Mengisi Saldo E-Dompet**

##### 5. Activity Diagram Mengelola Produk Penjualan

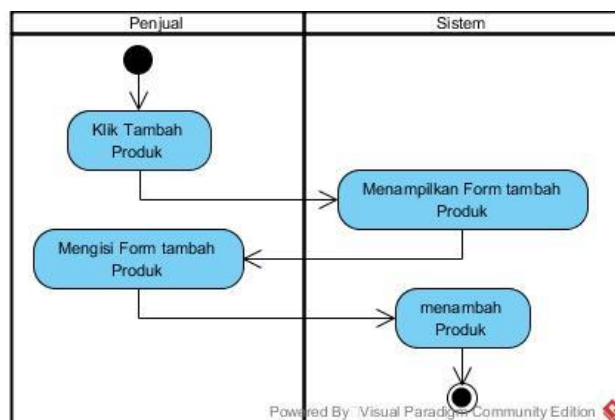
Ketika penjual ingin mengelola produk penjualan, Penjual menekan menu kelola produk. Sistem akan menampilkan opsi kelola produk. Berikut ini Activity Diagram Mengelola Produk Penjualan dapat dilihat pada Gambar 5.5.



**Gambar 5.5 Activity Diagram Mengelola Produk Penjualan**

##### 6. Activity Diagram Tambah Produk

Setelah memasuki menu kelola produk penjual menekan tombol tambah produk untuk menambah produk penjualan. Sistem akan menampilkan form tambah produk. Pembeli mengisi form tambah produk. Sistem menambah produk penjualan. Berikut ini Activity Diagram Tambah Produk dapat dilihat pada Gambar 5.6.



**Gambar 5.6 Activity Diagram Tambah Produk**

### 7. Activity Diagram Menghapus Produk

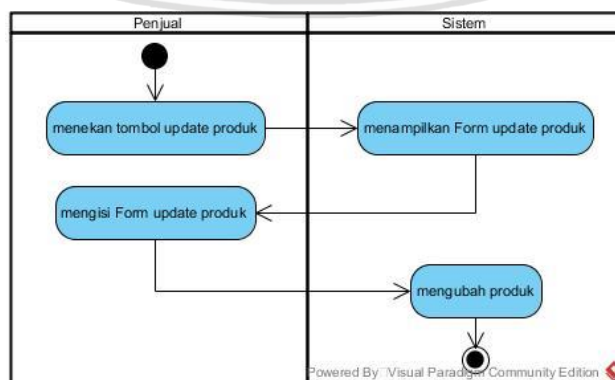
Setelah memasuki menu kelola produk penjual menekan tombol “x” untuk menghapus produk penjualan. Sistem akan menghapus produk penjualan. Berikut ini *Activity Diagram* Hapus Produk dapat dilihat pada Gambar 5.7.



**Gambar 5.7 Activity Diagram Menghapus Produk**

### 8. Activity Diagram Update Produk

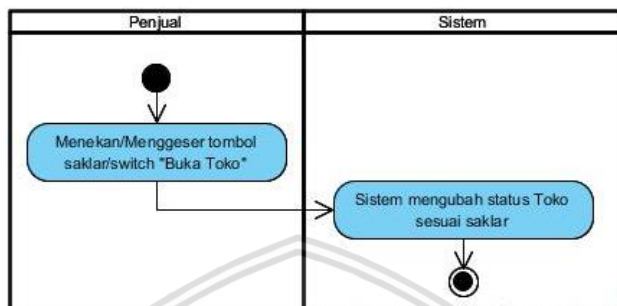
Setelah memasuki menu kelola produk penjual menekan tombol ubah stok. Sistem akan menampilkan form update stok produk. Penjual mengisi form update stok produk. Sistem mengubah stok produk. Berikut ini *Activity Diagram* Update Produk dapat dilihat pada Gambar 5.8.



**Gambar 5.8 Activity Diagram Update Produk**

9. *Activity Diagram* Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi)

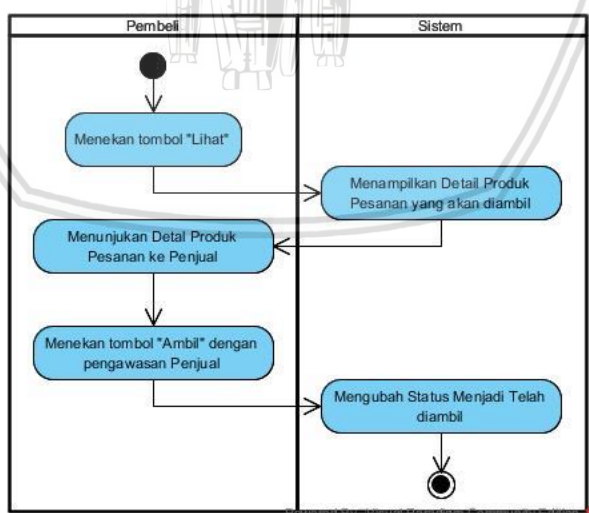
Pada menu kelola produk penjual menekan tombol saklar/switch buka toko. Sistem otomatis mengubah status toko sesuai saklar (jika saklar mati maka toko tutup dan jika saklar hidup maka toko buka). Berikut ini *Activity Diagram* Edit Status Toko dapat dilihat pada Gambar 5.9.



**Gambar 5.9** *Activity Diagram* Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi)

10. *Activity Diagram* Edit Status Pesanan diambil (tambahan pada iterasi ke-1 fase konstruksi)

Pada menu Riwayat Pesanan jika terdapat pesanan dengan status telah selesai pembeli menekan tombol "Lihat". Sistem akan menampilkan detail produk pesanan yang ingin diambil. Pembeli lalu melihatkan detail produk pesanan ke penjual dan menekan tombol "ambil" dengan pengawasan penjual. Sistem akan mengubah status produk pesanan menjadi telah diambil. Berikut ini *Activity Diagram* Edit Status Pesanan diambil dapat dilihat pada Gambar 5.10.

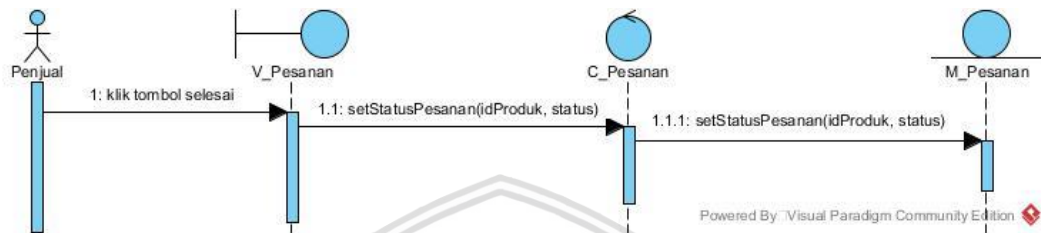


**Gambar 5.10** *Activity Diagram* Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi)



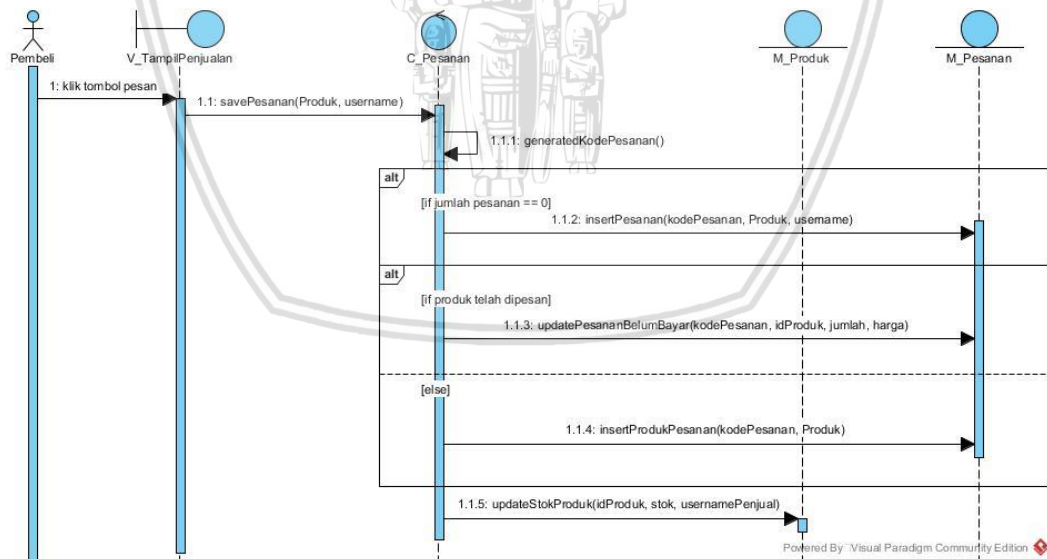
## 5.2 Sequence Diagram

Perancangan *sequence diagram* dibuat berdasarkan pada masing-masing *activity diagram* yang telah didefinisikan pada saat proses sebelumnya. Perancangan *sequence diagram* digunakan untuk melihat bagaimana sistem bekerja secara detail. Berikut ini masing-masing *sequence diagram* yang dapat dilihat pada gambar dibawah.



**Gambar 5.11 Sequence Diagram Edit Status Pesanan**

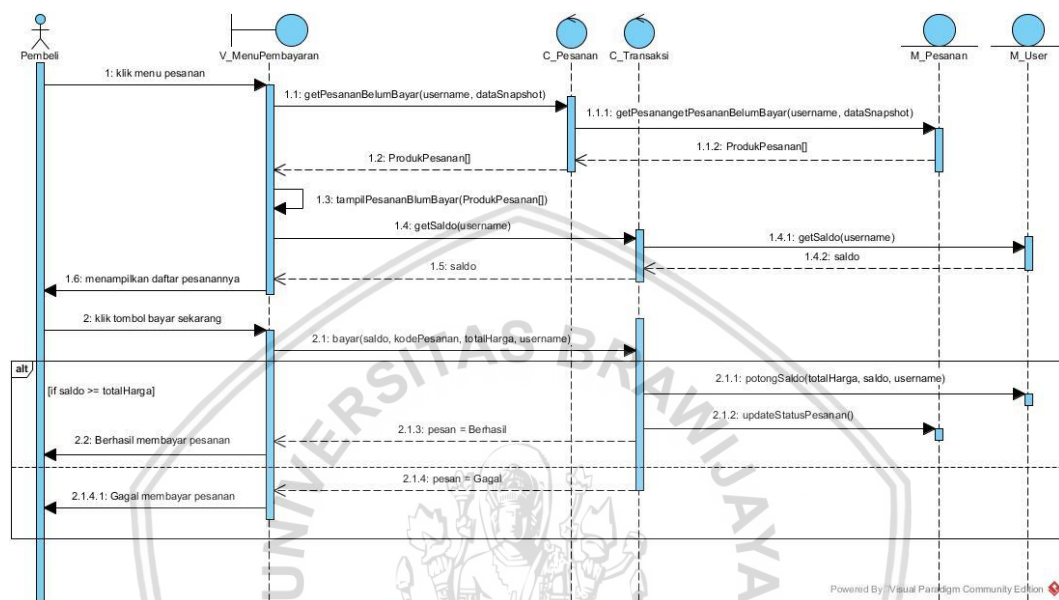
Gambar 5.11 menjelaskan alur proses sistem ketika penjual melakukan Edit Status Pesanan. Proses pertama dimulai ketika Penjual menekan tombol selesai pada tampilan V\_PesananPenjual, lalu dari V\_PesananPenjual akan memanggil method/fungsi setStatusPesanan() yang dimiliki oleh C\_Pesanan, setelah itu C\_Pesanan akan memanggil fungsi setStatusPesanan() yang dimiliki oleh M\_Pesanan untuk mengubah status pesanan menjadi selesai.



**Gambar 5.12 Sequence Diagram Memesan Produk Penjualan**

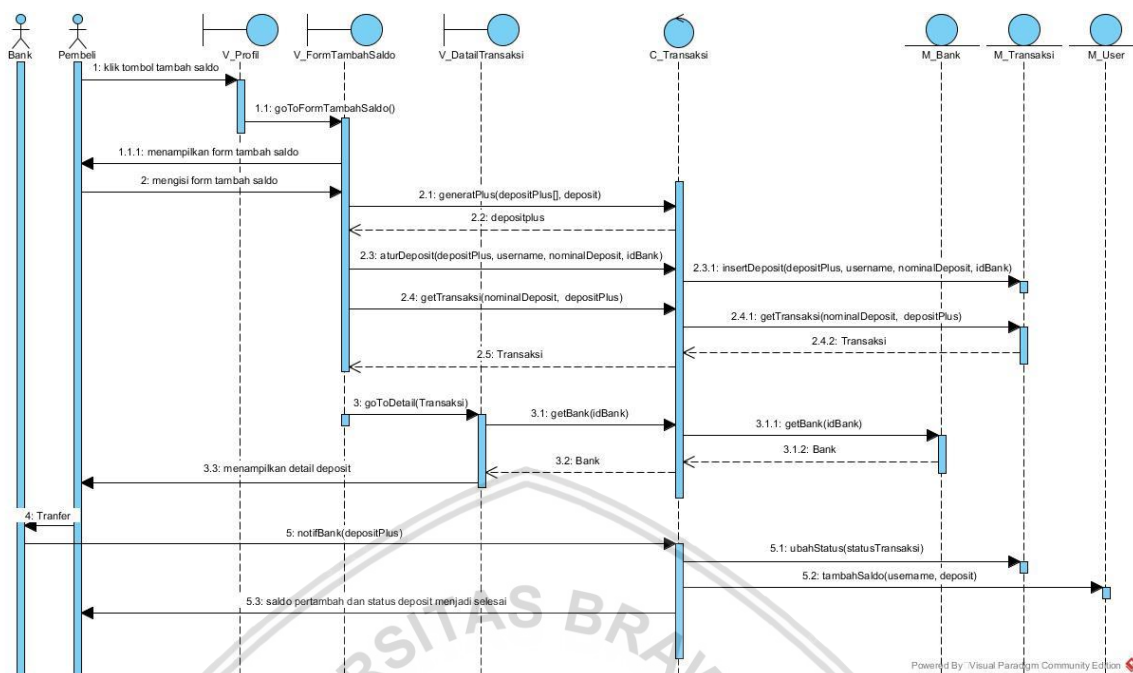
Gambar 5.12 menjelaskan alur proses sistem ketika pembeli melakukan pemesanan produk penjualan. Proses pertama dimulai ketika pembeli menekan tombol pesan/add pada V\_TampilPenjualan, lalu V\_TampilPenjualan akan memanggil fungsi savePesanan() yang dimiliki oleh C\_Pesanan untuk melakukan penyimpanan produk pesanan, setelah itu akan dilakukan generateKodePesanan()

untuk mendapatkan Kode Pesanan. Sebelum melakukan pemanggilan fungsi, dilakukan pengecekan kondisi jika jumlah pesanan = 0 maka akan memanggil fungsi insertPesanan(), jika produk telah dipesan maka akan memanggil updatePesananBelumBayar() dan selain itu maka akan memanggil insertProdukPesanan(). Selanjutnya akan memanggil fungsi updateStokProduk() untuk mengurangi stok produk yang dipesan.



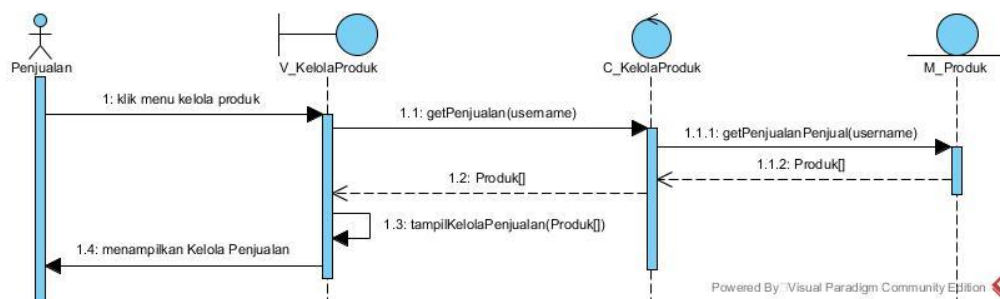
Gambar 5.13 Sequence Diagram Membayar Pesanan

Gambar 5.13 menjelaskan alur proses sistem ketika pembeli membayar pesanan. Proses pertama dimulai ketika pembeli menekan tombol pesan sekarang lalu dari V\_MenuPembayaran akan memanggil `getPesananBelumBayar()` yang dimiliki oleh C\_Pesanan dan fungsi tersebut akan memanggil `getPesananBelumBayar()` pada M\_Pesanan untuk mendapatkan data pesanan yang belum dibayar lalu dikembalikan nilainya ke fungsi sebelumnya. Setelah mendapatkan data pesanan yang belum dibayar V\_MenuPembayaran menjalankan fungsi `tampilPesananBlumBayar()` lalu memanggil `getSaldo()` pada C\_Transaksi dan fungsi tersebut akan memanggil `getSaldo()` pada M\_User untuk mendapatkan data saldo pembeli dan mengembalikan nilainya ke fungsi sebelumnya. Ketika pembeli menekan bayar sekarang V\_MenuPembayaran akan memanggil fungsi `bayar()` lalu didalam fungsi tersebut terdapat pengecekan kondisi jika `saldo >= totalHarga` maka akan memanggil fungsi `potongSaldo()` pada M\_User dan `updateStatusPesanan()` pada M\_Pesanan lalu mengembalikan nilai *true*/berhasil, dan jika tidak akan mengembalikan nilai *false*/gagal.



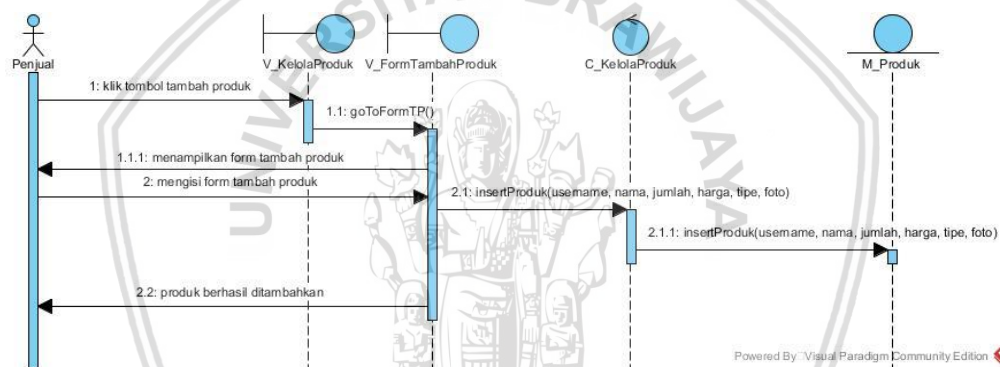
Gambar 5.14 Sequence Diagram Mengisi Saldo E-Dompet

Gambar 5.14 menjelaskan alur proses sistem ketika pembeli mengisi saldo E-Dompet. Proses pertama dimulai ketika pembeli menekan tombol tambah Saldo dari V\_Profil lalu V\_Profil menjalankan fungsi `gotoFormTambahSaldo()` untuk berpindah ke V\_FormTambahSaldo. Ketika pembeli selesai mengisi form dan menekan tombol Tambahkan maka V\_FormTambahSaldo akan memanggil `generatePlus()` untuk mengkalkulasi jumlah nominal deposit dengan 3 digit acak dan mengembalikan nilai berupa `depositPlus`, lalu memanggil fungsi `aturDeposit()` pada C\_Transaksi dan pada fungsi tersebut memanggil `insertDeposit()` untuk melakukan penambahan data transaksi deposit dengan status belum dibayar, setelah itu dari V\_FormTambahSaldo memanggil fungsi `getTransaksi()` pada C\_Transaksi dan pada fungsi tersebut memanggil `getTransaksi()` pada M\_Transaksi untuk mendapatkan data Transaksi yang baru saja ditambahkan dan mengembalikan nilainya pada fungsi sebelumnya. Setelah data Transaksi telah didapat V\_FormTambahSaldo akan memanggil fungsi `gotoDetail` untuk berpindah ketampilan ke V\_DetailTransaksi dengan membawa data Transaksi tersebut. V\_DetailTransaksi akan memanggil `getBank()` pada C\_Transaksi dan pada fungsi tersebut memanggil `getBank()` pada M\_Bank untuk mendapatkan data berupa Bank dan mengembalikan nilainya pada fungsi sebelumnya. Setelah itu bank akan memanggil `notifBank()` jika pembeli berhasil melakukan *transfer* ke bank tujuan, setelah itu pada fungsi `notifBank()` akan memanggil fungsi `ubahStatus()` pada M\_Transaksi dan `tambahSaldo` pada M\_User.



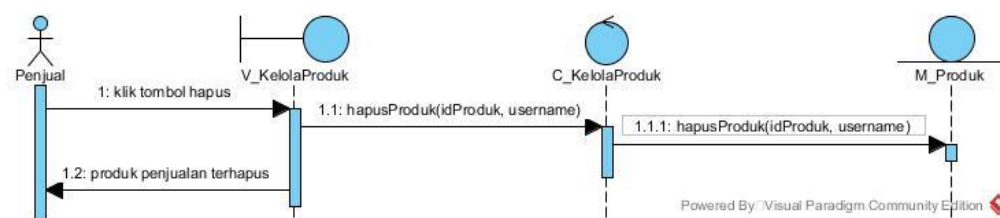
**Gambar 5.15 Sequence Diagram Mengelola Produk Penjualan**

Gambar 5.15 menjelaskan alur proses sistem ketika penjual ingin melakukan kelola Produk. Proses pertama dimulai ketika penjual menekan menu Kelola produk lalu V\_KelolaProduk memanggil getPenjualan() pada C\_KelolaProduk dan pada fungsi tersebut memanggil getPenjualan pada M\_Produk untuk mendapatkan data Produk lalu dikembalikan nilainya ke fungsi sebelumnya. Setelah itu V\_KelolaProduk menjalankan fungsi tampilKelolaPenjualan().



**Gambar 5.16 Sequence Diagram Tambah Produk**

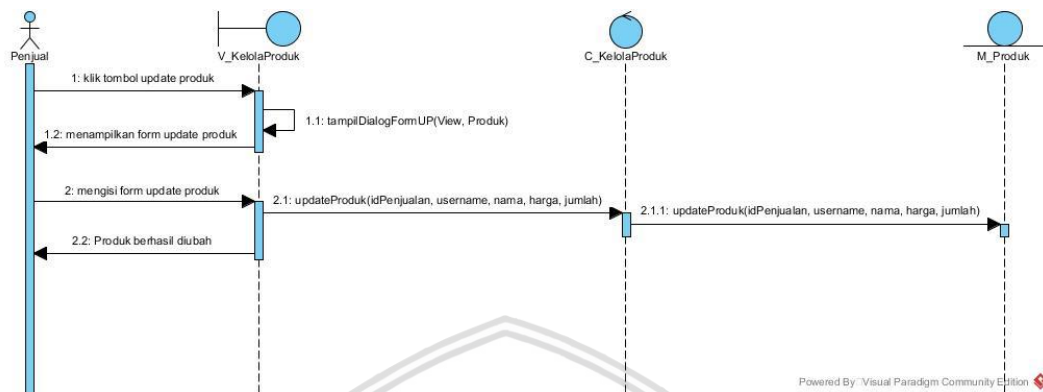
Gambar 5.16 menjelaskan alur proses sistem ketika penjual melakukan tambah produk. Proses pertama dimulai ketika penjual menekan tombol tambah produk dari V\_KelolaProduk lalu V\_KelolaProduk menjalankan fungsi goToFormTP() untuk berpindah ke V\_FormTambahProduk. Ketika penjual selesai mengisi form dan menekan tombol Tambahkan maka V\_KelolaProduk akan memanggil insertProduk() pada C\_KelolaProduk dan pada fungsi tersebut memanggil insertProduk() pada M\_Produk untuk melakukan penambahan data produk.



**Gambar 5.17 Sequence Diagram Hapus Produk**

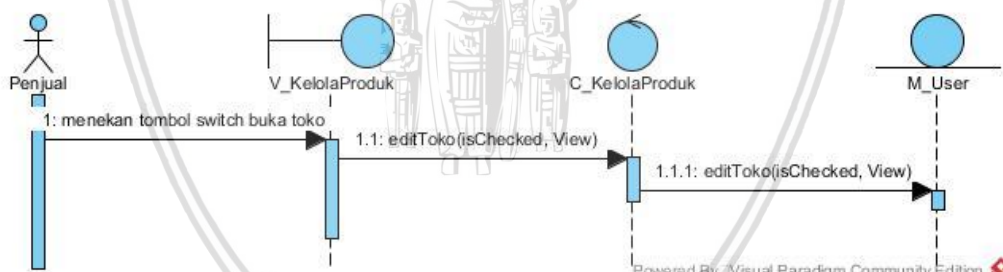
Gambar 5.17 menjelaskan alur proses sistem ketika penjual melakukan hapus produk. Proses pertama dimulai ketika penjual menekan tombol hapus produk

dari V\_KelolaProduk lalu V\_KelolaProduk menjalankan fungsi hapusProduk() pada C\_KelolaProduk dan pada fungsi tersebut akan memanggil hapusProduk() pada M\_Produk untuk menghapus data produk.



**Gambar 5.18 Sequence Diagram Update Produk**

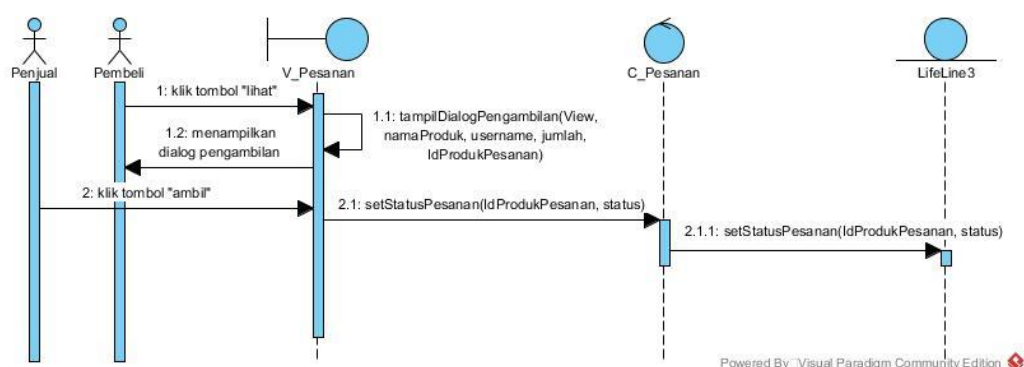
Gambar 5.18 menjelaskan alur proses sistem ketika penjual melakukan *update* produk. Proses pertama dimulai ketika penjual menekan tombol *edit* produk dari V\_KelolaProduk lalu V\_KelolaProduk menjalankan fungsi tampilDialogFormUP() untuk menampilkan dialog form update produk. Ketika penjual selesai mengisi form dan menekan tombol Selesai maka V\_KelolaProduk akan memanggil updateProduk() pada C\_KelolaProduk dan pada fungsi tersebut memanggil updateProduk() pada M\_Produk untuk melakukan *update* data produk.



**Gambar 5.19 Sequence Diagram Edit Status Toko (tambahan pada iterasi ke-1 fase konstruksi)**

Gambar 5.19 menjelaskan alur proses sistem ketika penjual ingin membuka stan mereka. Proses pertama dimulai ketika penjual menekan/menggeser tombol *switch* "buka toko" V\_KelolaProduk, lalu V\_KelolaProduk menjalankan fungsi editToko() untuk mengubah status toko sesuai parameter isChecked yang ada pada C\_KelolaProduk. Pada C\_KelolaProduk akan memanggil fungsi editToko() pada M\_User. Fungsi editToko() pada M\_User akan melakukan perubahan status toko sesuai parameter isChecked.





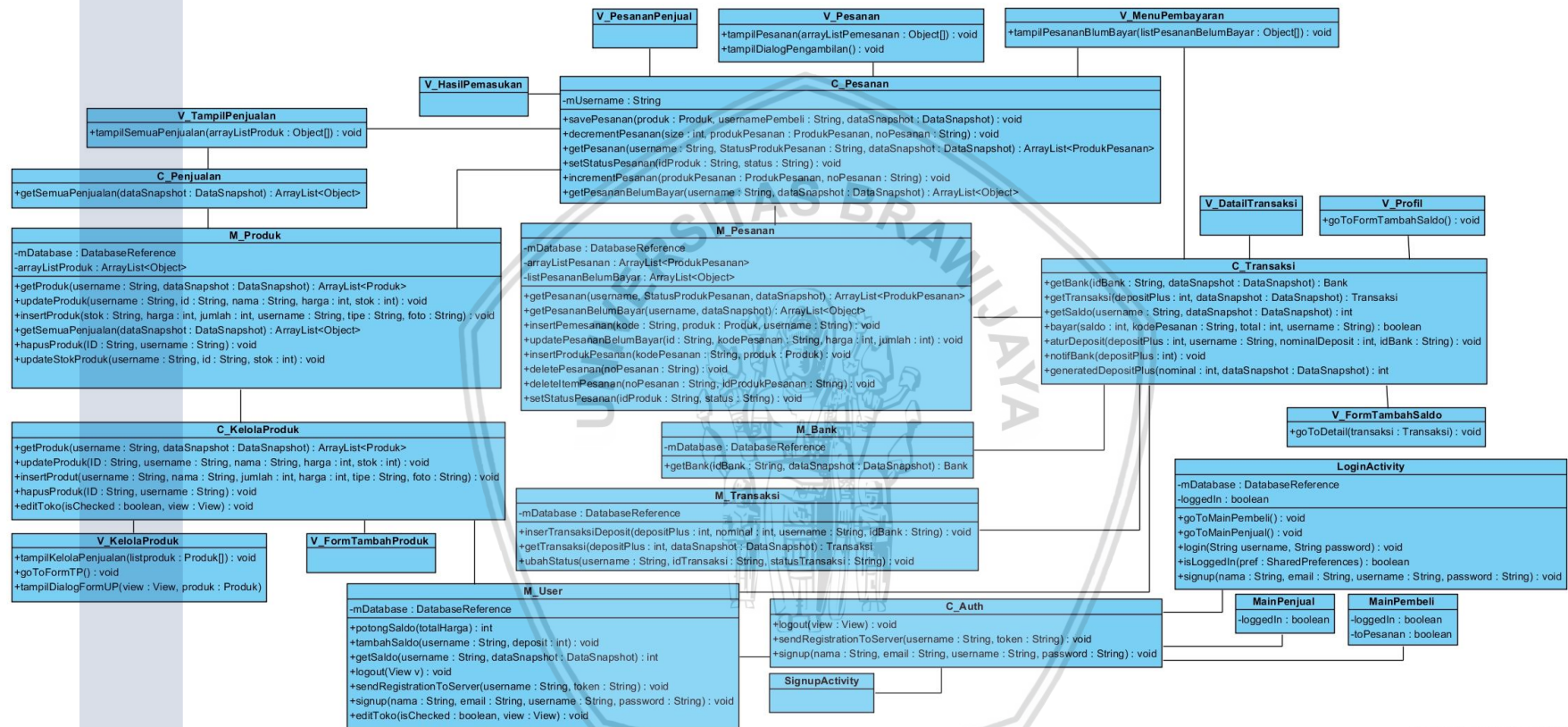
**Gambar 5.20 Sequence Diagram Edit Status Pesanan Diambil (tambahan pada iterasi ke-1 fase konstruksi)**

Gambar 5.20 menjelaskan alur proses sistem ketika pembeli menekan tombol “lihat” untuk menunjukkan bukti pemesanan ke penjual. Proses pertama dimulai ketika pembeli menekan tombol “Lihat” pada V\_Pesanan, lalu V\_Pesanan akan memanggil fungsi tampilDialogPengambilan() pada dirinya untuk menampilkan dialog detail produk pesanan. Setelah itu pembeli akan memperlihatkan detail tersebut ke penjual lalu penjual menekan tombol “ambil” pada V\_Pesanan, setelah itu V\_Pesanan akan memanggil fungsi setStatusPesanan() pada C\_Pesanan. Pada C\_Pesanan akan memanggil fungsi tampilDialogPengambilan() yang berada pada M\_Pesanan. Fungsi tampilDialogPengambilan() pada M\_Pesanan akan melakukan perubahan status produk pesanan menjadi diambil.

### 5.3 Class Diagram

*Class diagram* merepresentasikan class, atribut, fungsi dan hubungan antar class pada suatu perangkat lunak. *Class diagram* digunakan untuk memisah suatu fungsi ke beberapa class yang memiliki hubungan dengan fungsi tersebut. Berikut ini class diagram sistem informasi penjualan dan pembelian pada kantin filkom dapat dilihat pada Gambar 5.21.

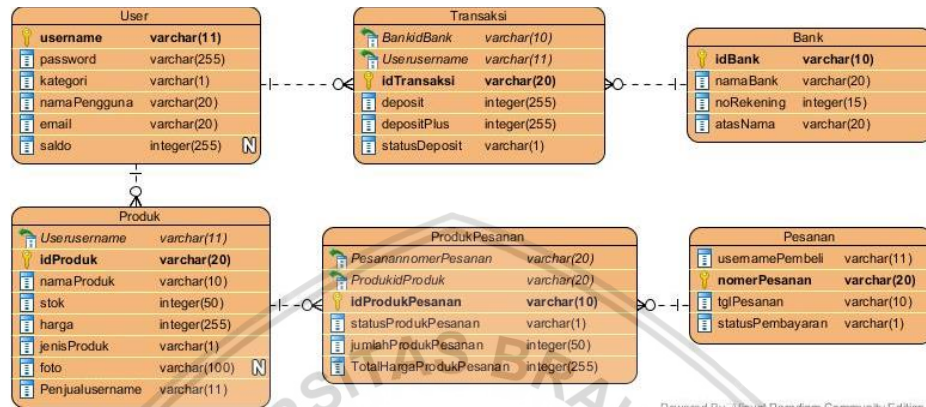




Gambar 5.21 Class Diagram Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom

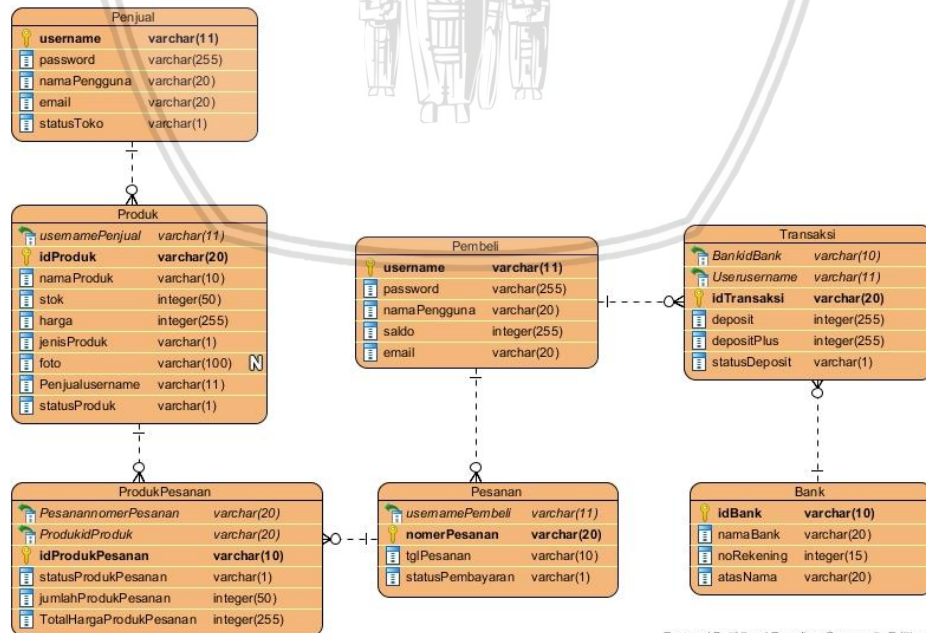
## 5.4 Physical Data Model

Perancangan ini digunakan untuk membuat rancangan bagaimana data direkap dalam sistem untuk mendukung proses pemesanan dan penjualan. Berdasarkan *class diagram* yang telah dimodelkan sebelumnya, maka diperoleh rancangan basis data yang direpresentasikan dalam bentuk physical data model seperti pada Gambar 5.22.



Gambar 5.22 Physical Data Model Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom

Pada awal fase konstruksi terdapat perubahan pada perancangan *Physical Data Model* dimana kelas User dipecah menjadi dua kelas yang berbeda yaitu Pembeli dan Penjual karena terdapat atribut yang berbeda antara pembeli dan penjual. Berikut perubahan perancangan *Physical Data Model* yang dapat dilihat pada Gambar 5.23.



Gambar 5.23 Physical Data Model Sistem Informasi Penjualan dan Pembelian pada Kantin Filkom (Awal Fase Construction)

## 5.5 Perancangan Antarmuka Pengguna (UI)

Pada tahap perancangan antarmuka pengguna ini merupakan proses pembuatan tampilan sistem yang akan dibangun agar pengguna dapat berinteraksi dengan sistem informasi pemesanan dan penjualan pada kantin filkom. Antarmuka pada sistem pemesanan dan penjualan ini terdapat 2 kategori yaitu Antarmuka Pembeli dan Antarmuka Penjual.

### Antarmuka Penjual

Perancangan pada antarmuka penjual digunakan untuk mengelola produk penjualan dan pesanan dari pembeli. Antarmuka ini juga menampilkan dapat menampilkan hasil pemasukkan dari semua penjualan yang telah dilakukan. Berikut adalah rancangan antarmuka untuk tampilan pembeli yang ada pada sistem yang dapat dilihat pada Gambar 5.24 sampai Gambar 5.27.

#### 1. Perancangan Antarmuka Pesanan

Pesanan	
<input type="checkbox"/>	Nama Pemesan Jumlah <input type="button" value="Selesai"/>
<input type="checkbox"/>	Nama Pemesan Jumlah <input type="button" value="Selesai"/>
Pesanan Selesai	
<input type="checkbox"/>	Nama Pemesan Jumlah <input type="button" value="Diambil"/>
Pesanan	Kelola Produk Hasil

**Gambar 5.24 Tampilan Pesanan**

Antarmuka pada Gambar 5.24 ini digunakan oleh penjual untuk melihat daftar pesanan yang ada. Terdapat informasi produk pesanan yang terbagi menjadi dua bagian yaitu pesanan baru dan pesanan yang belum diambil.

## 2. Perancangan Antarmuka Kelola Produk

The interface is titled 'Kelola Produk'. It displays a list of three products. Each product entry includes a placeholder image (a square with an 'X'), the product name, price, stock level, and product type. The first two products have a 'Ubah Stok' button, while the third has a 'Simpan' button. At the bottom, there is a navigation bar with a '+' icon and three tabs: 'Pesanan', 'Kelola Produk', and 'Hasil'.

**Gambar 5.25 Tampilan Kelola Produk**

Antarmuka pada Gambar 5.25 ini digunakan penjual untuk mengelola produk penjualan yang akan dijual. Terdapat informasi produk penjualan dan tombol-tombol untuk melakukan kelola produk seperti tambah, hapus, dan ubah stok.

## 3. Perancangan Antarmuka Tambah Produk

The interface is titled 'Tambah Produk'. It contains a form for adding a new product. The form includes input fields for 'Nama' (Name), 'Harga' (Price), 'Tipe produk' (Product type), and 'Stok' (Stock). There is also a placeholder image for the product photo. At the bottom of the form is a 'Tambahkan' button.

**Gambar 5.26 Tampilan Tambah Produk**

Antarmuka pada Gambar 5.26 ini digunakan penjual untuk menambahkan produk penjualan. Terdapat form pengisian untuk melakukan penambahan produk, inputan tersebut berupa nama, harga, tipe, stok, dan foto produk.

#### 4. Perancangan Antarmuka Hasil Penjualan

Hasil Penjualan	
Total Pemasukan	Rp 1.000.000
Total Pesanan	50 Pesanan
Pesanan	Kelola Produk
Hasil	

**Gambar 5.27 Tampilan Hasil Penjualan**

Antarmuka pada Gambar 5.27 ini digunakan penjual untuk dapat melihat hasil penjualan yang telah dilakukan. Terdapat informasi hasil penjualan berupa total pemasukan dan total pesanan yang telah selesai dilakukan.

#### Antarmuka Pembeli

Perancangan pada antarmuka pembeli digunakan untuk melakukan pemesanan suatu produk kantin seperti makanan dan minuman. Antarmuka ini juga menampilkan profil sang pembeli dimana terdapat informasi mengenai saldo pembeli. Berikut adalah rancangan antarmuka untuk tampilan pembeli yang ada pada sistem yang dapat dilihat pada Gambar 5.28 sampai Gambar 5.32.

#### 1. Perancangan Antarmuka Menu Penjualan

Menu Penjualan	
Nama Penjual	
	Nama Harga Tipe produk ADD
	Nama Harga Tipe produk 3
Nama Penjual	
	Nama Harga ADD
Total Harga: Rp 50000	
Pesan Sekarang	
Menu Penjualan	Pesanan
Profil saya	

**Gambar 5.28 Tampilan Menu Penjualan**



Antarmuka pada Gambar 5.28 ini digunakan oleh pembeli untuk melihat daftar menu penjualan. Terdapat informasi mengenai daftar produk penjualan mulai dari foto, nama, harga, dan tipe produk beserta nama penjual/kantin. Terdapat pula informasi total harga yang dipesan ketika telah memulai pemesanan.

## 2. Perancangan Antarmuka Pembayaran

Pembayaran	
	Nama Harga Tipe produk 3
	Nama Harga Tipe produk 3
<p>Total Harga: Rp 50000</p> <p>Saldo anda: Rp 100000</p> <p>Sisa Saldo: Rp 50000</p> <p>Bayar Sekarang</p>	

**Gambar 5.29 Tampilan Pembayaran**

Antarmuka pada Gambar 5.29 ini digunakan oleh pembeli untuk melakukan pembayaran pesanan. Terdapat informasi produk penjualan dan kalkulasi pembayaran seperti total harga, saldo, dan sisa saldo.

## 3. Perancangan Antarmuka Riwayat Pemesanan

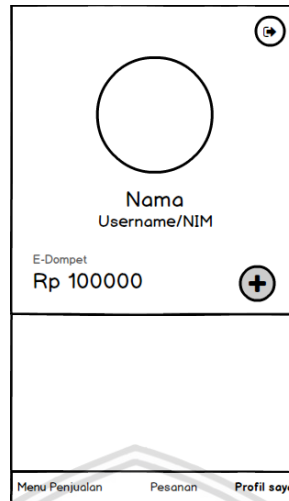
Riwayat Pemesanan	
Tanggal: 25 Sep 1017	
	Nama Jumlah Status
	Nama Jumlah Status
Tanggal: 15 Sep 1017	
	Nama Jumlah Status
<p>Menu Penjualan   Pesanan   Profil saya</p>	

**Gambar 5.30 Tampilan Riwayat Pemesanan**

Antarmuka pada Gambar 5.30 ini digunakan pembeli untuk melihat daftar riwayat pemesanan yang telah dilakukan. Terdapat informasi produk yang dipesan mulai dari nama, jumlah, dan status produk pesanan.



#### 4. Perancangan Antarmuka Profil Akun

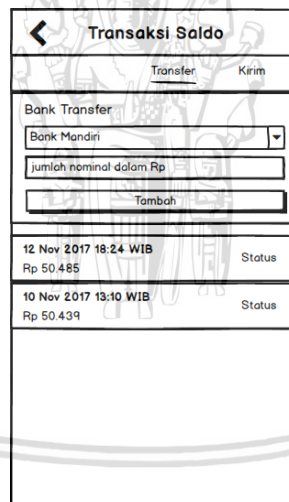


The mockup shows a vertical rectangular interface. At the top right is a camera icon. Below it is a large circular placeholder for a profile picture. Under the picture, the text 'Nama' and 'Username/NIM' is displayed. Below that, 'E-Dompet' and 'Rp 100000' is shown next to a plus icon. A large empty rectangular box is below the balance. At the bottom is a navigation bar with three items: 'Menu Penjualan', 'Pesanan', and 'Profil saya'.

**Gambar 5.31 Tampilan Profil Akun**

Antarmuka pada Gambar 5.31 ini digunakan untuk melihat profil pembeli dan melihat saldo e-dompet yang dimiliki. Terdapat informasi pembeli berupa foto, nama, username/nim, dan saldo pembeli.

#### 5. Perancangan Antarmuka Tambah Saldo



The mockup is titled 'Transaksi Saldo' with a back arrow. It has two tabs: 'Transfer' (selected) and 'Kirim'. Under 'Bank Transfer', there is a dropdown menu showing 'Bank Mandiri', a text input for 'jumlah nominal dalam Rp', and a 'Tambah' button. Below this is a list of transactions with two rows: '12 Nov 2017 18:24 WIB Rp 50.485 Status' and '10 Nov 2017 13:10 WIB Rp 50.439 Status'. A large empty box is at the bottom.

**Gambar 5.32 Tampilan Tambah Saldo**

Antarmuka pada Gambar 5.32 ini digunakan ketika pembeli ingin melakukan penambahan/pengisian saldo e-dompet. Terdapat form pengisian untuk melakukan pengisian saldo, inputan tersebut berupa bank tujuan dan nominal saldo yang ingin diisi.

## BAB 6 IMPLEMENTASI SISTEM

Bab ini membahas mengenai proses implementasi dari perancangan sistem yang telah dibuat dalam membangun sebuah sistem informasi pemesanan dan penjualan kantin filkom. Dalam bab ini akan sedikit membahas Fase *Elaboration* dan akan banyak membahas pada Fase *Construction* pada *RUP*. Pada bab ini terdapat tahapan *Implementation* dan iterasinya. Tahapan *implementasi* adalah proses menerjemahkan desain ke dalam kode sistem. Implementasi bertujuan untuk melakukan pengelompokkan kode, kelas dan objek, menguji komponen yang dikembangkan, dan mengintegrasikan hasil dari pengembangan subsistem menjadi sistem yang utuh.

### 6.1 Iterasi Ke-1 Fase *Elaboration*

Iterasi Ke-1 pada fase elaborasi juga sudah memulai tahap awal dalam implementasi sistem informasi pemesanan dan penjualan kantin filkom.

#### 6.1.1 Spesifikasi Sistem

Spesifikasi sistem berdasarkan rancangan, diimplementasikan pada 2 jenis spesifikasi, yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

##### Spesifikasi Perangkat Keras

Spesifikasi Perangkat Keras yang digunakan dalam implementasi meliputi perangkat keras komputer, perangkat bergerak, dan server. Perangkat keras komputer digunakan untuk melakukan pengkodean program, pada perangkat bergerak digunakan untuk proses *debugging* dan pada perangkat keras Server sendiri menggunakan *Firebase Console*. Penjelasan tabel spesifikasi akan dijelaskan pada Tabel 6.1, Tabel 6.2 dan Tabel 6.3.

Tabel 6.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>Merk</i>	Acer Aspire 4752
<i>Operating System</i>	Windows 10 Enterprise 64-bit
<i>Processor</i>	Intel(R) Core(TM) i3-2330M 2.2GHz
<i>Graphic</i>	Intel(R) HD Graphic 3000
<i>RAM</i>	8GB

Tabel 6.2 Spesifikasi Perangkat Keras Perangkat Bergerak

Nama Komponen	Spesifikasi
<i>Merk</i>	Xiaomi Redmi 3
<i>Operating System</i>	Android 5.1.1 Lollipop

Nama Komponen	Spesifikasi
<i>Processor</i>	Octa-core Max 1.5GHz Cortex-A53
<i>Ghraphic</i>	Qualcomm Adreno 405
<i>RAM</i>	2GB

Tabel 6.3 Spesifikasi Perangkat Keras Server

Nama Komponen	Spesifikasi
<i>Server Name</i>	Firebase Console
<i>Services</i>	Realtime Database

### Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk pengembangan sistem informasi pemesanan dan penjualan pada kantin filkom dijelaskan pada Tabel 6.4.

Tabel 6.4 Spesifikasi Perangkat Lunak Pengembangan

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Enterprise 64-bit
<i>Programming Language</i>	Java Android
<i>Tools Development</i>	Android Studio
<i>Web Browser</i>	Mozilla Firefox

### 6.1.2 Batasan Implementasi

Program yang di buat memiliki beberapa Batasan dalam proses implementasi sistem informasi pemesanan dan penjualan pada kantin filkom, yaitu:

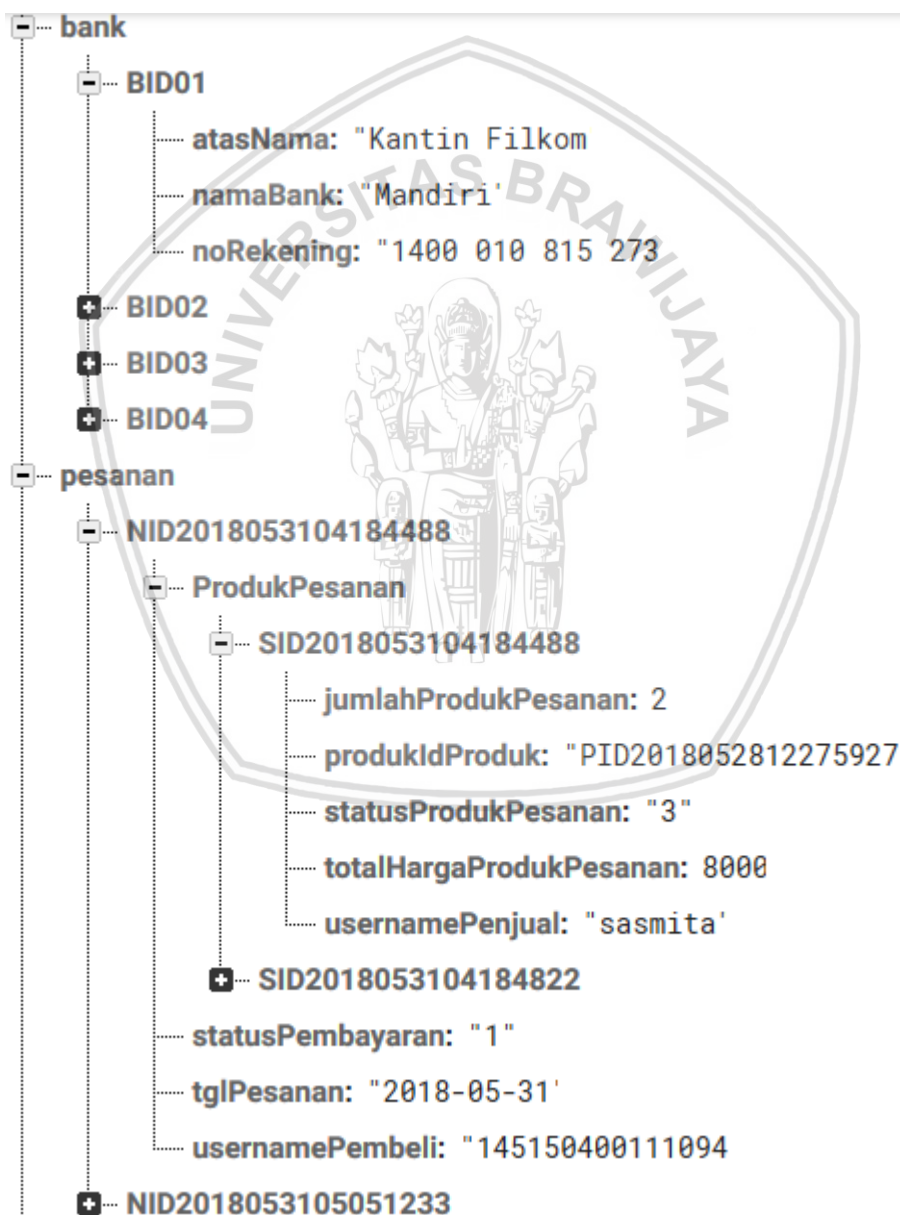
1. Aplikasi perangkat bergerak sistem informasi pemesanan dan penjualan pada kantin filkom hanya berjalan pada OS Android.
2. Aplikasi perangkat bergerak sistem informasi pemesanan dan penjualan pada kantin filkom dikembangkan dengan minimal spesifikasi OS android 4.1 Ice Cream Sandwich sampai dengan OS android terbaru.
3. Menggunakan firebase realtime database versi gratis sebagai server database, sehingga terdapat batasan pengguna 100 dan 1GB penyimpanan data.
4. Menggunakan Java Android sebagai bahasa pemrograman dan Android Studio sebagai *tools*-nya.

## 6.2 Fase Construction

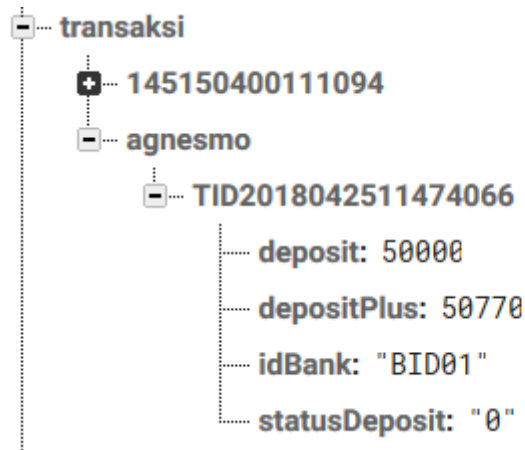
Awal pada fase konstruksi dibab ini mulai dengan melakukan tahap implementasi. tahap ini akan melakukan implementasi basis data, implementasi kelas, implementasi kode program, dan implementasi antarmuka pengguna.

### 6.2.1 Implementasi Basis Data

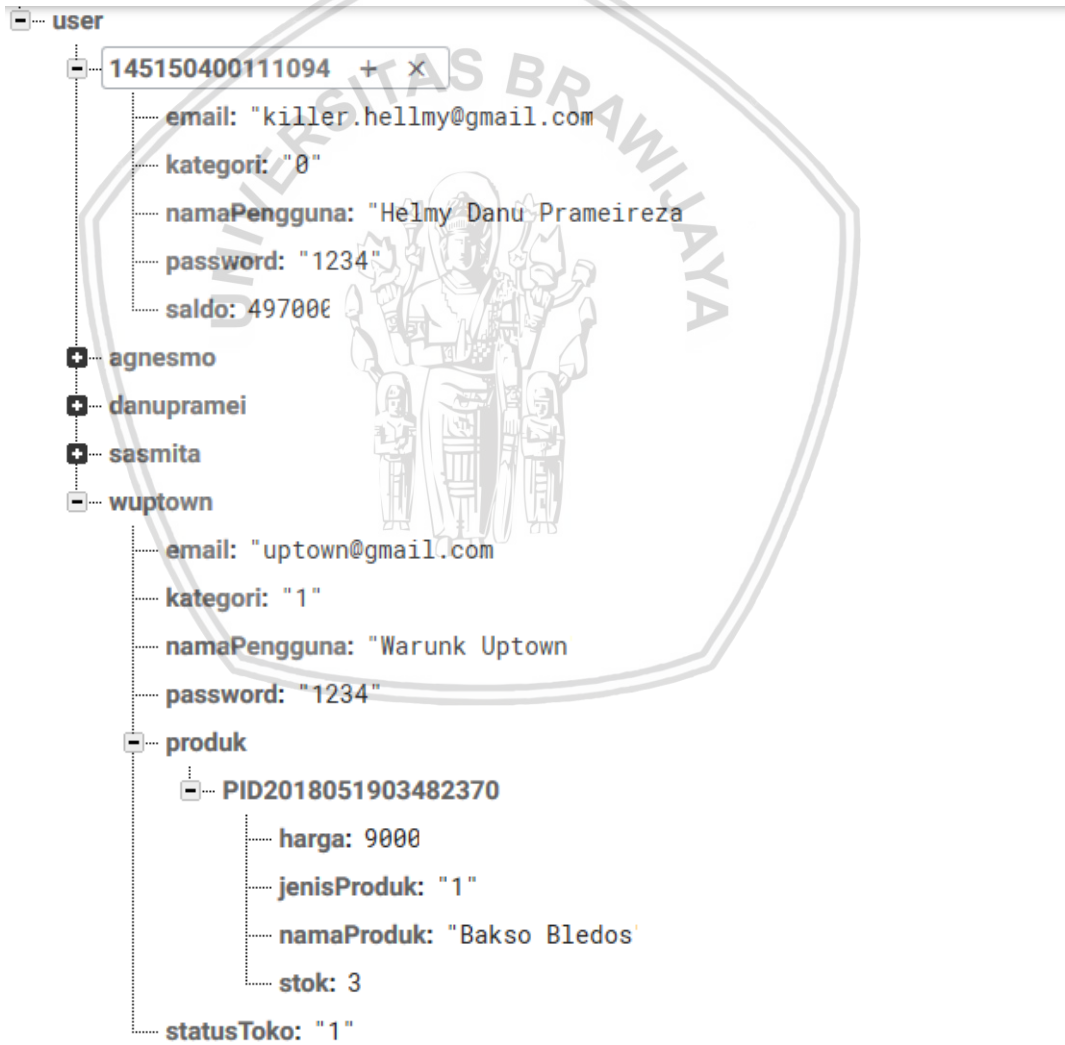
Implementasi basis data pada aplikasi perangkat bergerak sistem informasi pemesanan dan penjualan pada kantin filkom menggunakan firebase realtime database. Firebase *realtime database* menggunakan konsep No-SQL dan bentuk data yang menyerupai JSON, dengan konsep tersebut Firebase *database* dapat mengakses secara realtime. Berikut ini hasil dari implementasi basis data Firebase.



Gambar 6.1 Implementasi Basis Data Firebase (bank dan pesanan)



Gambar 6.2 Implementasi Basis Data Firebase (transaksi)



Gambar 6.3 Implementasi Basis Data Firebase (user)

### 6.2.2 Implementasi Kelas

Berdasarkan hasil rancangan aplikasi perangkat bergerak sistem informasi pemesanan dan penjualan pada kantin filkom terdapat beberapa kategori kelas yaitu *Model Class*, *View Class*, *Controller Class* dan juga *Physical Data Model Class*. Berikut masing-masing penjelasan dan implementasi kelas tersebut.

#### Implementasi Kelas *Model*

Kelas *model* adalah kelas yang berisikan fungsi pemanggilan data dari database server. Kelas *model* selalu berinteraksi dengan *database server*.

**Tabel 6.5 Berkas Kelas *Model***

No.	Nama Kelas	Nama Folder	Nama Layout
1	M_Bank	Model	-
2	M_Pesanan	Model	-
3	M_Produk	Model	-
4	M_Transaksi	Model	-
5	M_User	Model	-

Tabel 6.5 menjelaskan bahwa hasil implementasi kelas diagram terdapat berkas kelas Model dengan total 5 kelas didalamnya. Kelas tersebut berfungsi sebagai perantara dalam pengambilan data dari server basis data.

#### Implementasi Kelas *View*

Kelas *view* merupakan kelas yang merepresentasikan tampilan sebuah program. Kelas *view* selalu berinteraksi dengan *layout-layout* program dan ketika ingin meminta data, kelas *view* akan memanggil fungsi pada kelas *controller*.

**Tabel 6.6 Berkas Kelas *View***

No.	Nama Kelas	Nama Folder	Nama Layout
1	LoginActivity	-	activity_login.xml
2	SignupActivity	-	activity_signup.xml
3	MainPembeli	-	activity_main_pembeli.xml
4	MainPenjual	-	activity_main_pembeli.xml
5	V_TampilPenjualan	View	activity_menu_penjualan.xml
6	V_Pesanan	View	fragment_pesanan.xml
7	V_Profil	View	fragment_profil.xml



No.	Nama Kelas	Nama Folder	Nama Layout
8	V_FormTambahSaldo	View	activity_v_form_tambah_saldo.xml
9	V_MenuPembayaran	View	activity_menu_pembayaran.xml
10	V_PesananPenjual	View	fragment_pesanan_penjual.xml
11	V_KelolaProduk	View	fragment_kelola_produk.xml
12	V_HasilPemasukan	View	fragment_hasil.xml
13	V_FormTambahProduk	View	activity_v_form_tambah_produk.xml
14	V_DatailTransaksi	View	activity_v_detail_transaksi.xml

Tabel 6.6 menjelaskan bahwa hasil implementasi kelas diagram terdapat kelas yang berperan sebagai kelas View dengan total 14 kelas. Kelas *view* merupakan kelas yang merepresentasikan tampilan sebuah program. Kelas *view* selalu berinteraksi dengan *layout-layout* program dan ketika ingin meminta data, kelas *view* akan memanggil fungsi pada kelas *controller*.

#### Implementasi Kelas *Controller*

Kelas *controller* adalah kelas penghubung antara kelas *view* dan kelas *model*. Kelas *controller* juga merupakan kelas yang berisikan logika-logika pemrograman.

**Tabel 6.7 Berkas Kelas *Controller***

No.	Nama Kelas	Nama Folder	Nama Layout
1	C_KelolaProduk	Controller	-
2	C_Pesanan	Controller	-
3	C_Penjualan	Controller	-
4	C_Transaksi	Controller	-
5	C_Auth	Controller	-

Tabel 6.7 menjelaskan bahwa terdapat 5 kelas yang berperan sebagai *controller*. Kelas *controller* adalah kelas penghubung antara kelas *view* dan kelas *model*. Kelas *controller* juga merupakan kelas yang berisikan logika-logika pemrograman.

### Implementasi Kelas *Physical Data Model*

Kelas *Physical Data Model* adalah kelas yang memuat data fisik dalam database klien dan akan dipanggil untuk menyimpan atau menampilkan data dari database.

**Tabel 6.8 Berkas Kelas *Physical Data Model***

No.	Nama Kelas	Nama Folder	Nama Layout
1	Bank	DataModel	-
2	Pembeli	DataModel	-
3	Penjual	DataModel	-
4	Pesanan	DataModel	-
5	Produk	DataModel	-
6	ProdukPesanan	DataModel	-
7	Transaksi	DataModel	-

Tabel 6.8 menjelaskan bahwa terdapat 7 kelas yang berperan sebagai *physical data model*. Kelas *Physical Data Model* adalah kelas yang memuat data fisik dalam database klien dan akan dipanggil untuk menyimpan atau menampilkan data dari database. Kelas *Physical Data Model* akan dipanggil sebagai obyek data.

### 6.2.3 Implementasi Kode Program

Tahap implementasi Kode program merupakan tahap dimana melakukan penulisan kode fungsi program berdasarkan perancangan yang telah dibuat sebelumnya. Berikut hasil implementasi kode program yang telah dibuat.

#### 6.2.3.1 Kode Program Memesan Produk Penjualan

Pada fungsional memesan produk penjualan terdapat beberapa kode fungsi yang disematkan. Berikut tabel Kode Program yang dapat dilihat pada Tabel 6.9 sampai Tabel 6.14.

**Tabel 6.9 Kode Program Fungsi savePesanan()**

1	public class C_Pesanan {
2	public void savePesanan(Produk produk, String
3	usernamePembeli, DataSnapshot dataSnapshot){
4	if (produk.getStok() > 0){
5	for (DataSnapshot noteDataSnapshot :
6	dataSnapshot.getChildren()){
7	Pesanan pesanan =
8	noteDataSnapshot.getValue(Pesanan.class);
9	
10	pesanan.setNomerPesanan(noteDataSnapshot.getKey());
11	if
12	(!pesanan.getStatusPembayaran().equalsIgnoreCase("1") &&
13	pesanan.getUsernamePembeli().equalsIgnoreCase(usernamePemb
14	eli)){
15	

```

16         arrayListPemesanan.add(new
17 Pesanan(pesanan.getNomerPesanan(),
18 pesanan.getStatusPembayaran(),
19 pesanan.getUsernamePembeli(),pesanan.getTglPesanan()));
20         for (DataSnapshot
21 noteDataSnapshotChild :
22 noteDataSnapshot.child("ProdukPesanan").getChildren()){
23             ProdukPesanan produkPesanan =
24 noteDataSnapshotChild.getValue(ProdukPesanan.class);
25
26 produkPesanan.setIdProdukPesanan(noteDataSnapshotChild.get
27 Key());
28             arrayListPemesanan.add(new
29 ProdukPesanan(produkPesanan.getIdProdukPesanan(),
30 produkPesanan.getStatusProdukPesanan(),
31 produkPesanan.getJumlahProdukPesanan(),
32 produkPesanan.getTotalHargaProdukPesanan(),
33 produkPesanan.getProdukIdProduk()));
34
35 }
36
37 }
38
39 }
40
41 int count = 0;
42 String kodePesanan;
43 String usernamePenjual = produk.getUsername();
44 String idProduk = produk.getIdProduk();
45 if (arrayListPemesanan.size() == 0) {
46     kodePesanan = generatedKodePesanan();
47     mPesanan.insertPemesanan(kodePesanan,
48 produk, usernamePembeli);
49 } else {
50     final int jumlah =
51 arrayListPemesanan.size();
52
53     kodePesanan = ((Pesanan)
54 arrayListPemesanan.get(0)).getNomerPesanan();
55     for (DataSnapshot noteDataSnapshot :
56 dataSnapshot.getChildren()) {
57         Pesanan pesanan =
58 noteDataSnapshot.getValue(Pesanan.class);
59         if
60 (!pesanan.getStatusPembayaran().equalsIgnoreCase("1") &&
61 pesanan.getUsernamePembeli().equalsIgnoreCase(usernamePemb
62 eli)){
63             for (DataSnapshot
64 noteDataSnapshotChild :
65 noteDataSnapshot.child("ProdukPesanan").getChildren()){
66                 ProdukPesanan produkPesanan =
67 noteDataSnapshotChild.getValue(ProdukPesanan.class);
68
69 produkPesanan.setIdProdukPesanan(noteDataSnapshotChild.get
70 Key());
71                 String IdAsli =
72 produk.getIdProduk();
73                 String cekId =
74 produkPesanan.getProdukIdProduk();

```

75	if
76	(IdAsli.equalsIgnoreCase(cekId)) {
77	String IdProdukPesanan =
78	produkPesanan.getIdProdukPesanan();
79	
80	int hargaAwal =
81	produk.getHarga();
82	int jumlahPesanan =
83	produkPesanan.getJumlahProdukPesanan()+1;
84	
85	mPesanan.updatePesananBelumBayar(IdProdukPesanan,
86	kodePesanan, hargaAwal*jumlahPesanan, jumlahPesanan);
87	
88	count++;
89	}
90	}
91	}
92	}
93	if (count == 0) {
94	
95	mPesanan.insertProdukPesanan(kodePesanan, produk);
96	}
97	}
98	mProduk.updateStokProduk(usernamePenjual,
99	idProduk, produk.getStok()-1);
100	}
101	}
102	}

Kode fungsi savePesanan() pada Tabel 6.9 bertugas sebagai menyimpan pesanan sesuai kondisi. fungsi savePesanan() terdapat beberapa pengkondisian yaitu jika stok produk kosong maka fungsi ini tidak dijalankan, dan jika stok produk tidak kosong maka akan dilakukan pengambilan data pesanan dan melakukan pengecekan lagi jika jumlah pesanan 0 maka akan memanggil fungsi generatedKodePesanan() untuk menginisialisasi kodePesanan dan setelah itu memanggil fungsi insertPesanan() untuk melakukan penyimpanan data pesanan ke database firebase. Namun jika jumlah pesanan tidak 0 maka dilakukan pengecekan lagi jika idProduk sama dengan idProdukPesanan maka akan melakukan kalkulasi ulang jumlah produk pesanan dan total harga dan memanggil fungsi updatePesananBelumBayar() untuk mengganti jumlah produk pesanan dan total harga. Selain itu jika kondisi tersebut tidak dijalankan maka akan memanggil fungsi insertProdukPesanan() untuk menyimpan data produk pesanan pada database firebase. Pada akhir kode fungsi ketika salah satu kondisi diatas dijalankan setelah itu akan memanggil fungsi updateStokProduk() untuk melakukan pengurangan stok produk penjualan sesuai dengan jumlah yang dipesan.

**Tabel 6.10 Kode Program Fungsi generatedKodePesanan()**

1	public class C_Pesanan {
2	public String generatedKodePesanan() {
3	String kodePesanan;
4	kodePesanan = "NID" + new
5	SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date());
6	return kodePesanan;

7	}
8	}

Kode fungsi `generatedKodePesanan()` pada Tabel 6.10 bertugas untuk menginisialisasi `kodePesanan` sesuai tanggal sampai milidetik saat ini, dengan format "NID"+"yyyyMMddHHmmssSS".

**Tabel 6.11 Kode Program Fungsi `insertPesanan()`**

1	<code>public class M_Pesanan {</code>
2	<code>    public void insertPemesanan(String kode, Produk</code>
3	<code>    produk, String username) {</code>
4	<code>        ProdukPesanan produkPesanan = new ProdukPesanan(</code>
5	<code>            "1",</code>
6	<code>            1,</code>
7	<code>            produk.getHarga(),</code>
8	<code>            produk.getIdProduk());</code>
9	
10	<code>        mDatabase =</code>
11	<code>        FirebaseDatabase.getInstance().getReference();</code>
12	<code>        String date = new SimpleDateFormat("yyyy-MM-</code>
13	<code>        dd").format(new Date());</code>
14	<code>        String idProdukPesanan = "SID"+new</code>
15	<code>        SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date());</code>
16	<code>        Pesanan pesanan = new Pesanan("0", username,</code>
17	<code>        date);</code>
18	
19	<code>        mDatabase.child("pesanan").child(kode).setValue(pesanan);</code>
20	
21	<code>        mDatabase.child("pesanan").child(kode).child("ProdukPesana</code>
22	<code>        n").child(idProdukPesanan).setValue(produkPesanan);</code>
23	
24	<code>        mDatabase.child("pesanan").child(kode).child("ProdukPesana</code>
25	<code>        n").child(idProdukPesanan).child("usernamePenjual").setVal</code>
26	<code>        ue(produk.getUsername());</code>
27	<code>    }</code>
28	<code>}</code>

Kode fungsi `insertPesanan()` pada Tabel 6.11 bertugas untuk melakukan penyimpanan data pesanan dan `produkPesanan` pada database firebase. Pertama melakukan penyimpanan data produk pesanan pada objek `ProdukPesanan`, lalu melakukan instansiasi database firebase. Pada baris selanjutnya melakukan inisialisasi `idProdukPesanan` sesuai tanggal sampai milidetik saat ini, dengan format "SID"+"yyyyMMddHHmmssSS", setelah itu melakukan penyimpanan data pesanan pada objek `Pesanan`. Setelah selesai mempersiapkan datanya maka selanjutnya melakukan penyimpanan objek `Pesanan` pada database firebase dengan *primary key* `kodePesanan` dan juga penyimpanan objek `ProdukPesanan` pada database firebase dengan *primary key* `idProdukPesanan`.

**Tabel 6.12 Kode Program Fungsi `updatePesananBelumBayar()`**

1	<code>public class M_Pesanan {</code>
2	<code>    public void updatePesananBelumBayar(String id, String</code>
3	<code>    kodePesanan, int harga, int jumlah){</code>
4	<code>        mDatabase =</code>
5	<code>        FirebaseDatabase.getInstance().getReference();</code>
6	
7	<code>        mDatabase.child("pesanan").child(kodePesanan).child("Produ</code>

8	kPesanan").child(id).child("totalHargaProdukPesanan").setV
9	alue(harga);
10	
11	mDatabase.child("pesanan").child(kodePesanan).child("Produ
12	kPesanan").child(id).child("jumlahProdukPesanan").setValue
13	(jumlah);
14	}
15	}

Kode fungsi `updatePesananBelumBayar()` pada Tabel 6.12 bertugas untuk mengganti data jumlah produk pesanan dan total harga produk pesanan. Pertama melakukan instansiasi database firebase, lalu melakukan penyimpanan data `totalHargaProdukPesanan` dengan total harga produk pesanan yang baru dan penyimpanan `jumlahProdukPesanan` dengan jumlah produk pesanan yang baru.

**Tabel 6.13 Kode Program Fungsi `insertProdukPesanan()`**

1	public class M_Pesanan {
2	public void insertProdukPesanan(String kodePesanan,
3	Produk produk){
4	ProdukPesanan produkPesanan = new ProdukPesanan(
5	"1",
6	1,
7	produk.getHarga(),
8	produk.getIdProduk());
9	
10	String idProdukPesanan = "SID"+new
11	SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date());
12	mDatabase =
13	FirebaseDatabase.getInstance().getReference();
14	
15	mDatabase.child("pesanan").child(kodePesanan).child("Produ
16	kPesanan").child(idProdukPesanan).setValue(produkPesanan);
17	
18	mDatabase.child("pesanan").child(kodePesanan).child("Produ
19	kPesanan").child(idProdukPesanan).child("usernamePenjual")
20	.setValue(produk.getUsername());
21	}
22	}

Kode fungsi `insertProdukPesanan()` pada Tabel 6.13 bertugas untuk menyimpan data produk pesanan pada database firebase. Pertama melakukan penyimpanan data produk pesanan pada objek `ProdukPesanan`, lalu melakukan inisialisasi `idProdukPesanan` sesuai tanggal sampai milidetik saat ini, dengan format "SID"+"yyyyMMddHHmmssSS". Pada baris selanjutnya melakukan instansiasi database firebase. Setelah selesai mempersiapkan datanya maka selanjutnya melakukan penyimpanan objek `ProdukPesanan` pada database firebase dengan *primary key* `idProdukPesanan`.

**Tabel 6.14 Kode Program Fungsi `updateStokProduk()`**

1	public class M_Produk {
2	public void updateStokProduk(String username, String
3	id, int stok){
4	mDatabase =
5	FirebaseDatabase.getInstance().getReference();
6	
7	



8	
9	mDatabase.child("user").child(username).child("produk").ch
10	ild(id).child("stok").setValue(stok);
	}
	}

Kode fungsi `updateStokProduk()` pada Tabel 6.14 bertugas untuk mengubah stok produk penjualan pada database firebase. Pertama melakukan instansiasi database firebase, selanjutnya melakukan penyimpanan data stok produk dengan stok yang telah berkurang.

### 6.2.3.2 Kode Program Membayar Pesanan

Pada fungsional membayar pesanan terdapat beberapa kode fungsi yang disematkan. Berikut tabel Kode Program yang dapat dilihat pada Tabel 6.15 sampai Tabel 6.19.

**Tabel 6.15 Kode Program Fungsi `getPesananBelumBayar()`**

1	public class M_Pesanan {
2	public ArrayList<Object> getPesananBelumBayar(String
3	username, DataSnapshot dataSnapshot) {
4	for (DataSnapshot noteDataSnapshot :
5	dataSnapshot.getChildren()) {
6	Pesanan pesanan =
7	noteDataSnapshot.getValue(Pesanan.class);
8	
9	pesanan.setNomerPesanan(noteDataSnapshot.getKey());
10	
11	if
12	(!pesanan.getStatusPembayaran().equalsIgnoreCase("1") &&
13	pesanan.getUsernamePembeli().equalsIgnoreCase(username)) {
14	for (DataSnapshot noteDataSnapshotChild :
15	noteDataSnapshot.child("ProdukPesanan").getChildren()) {
16	ProdukPesanan produk =
17	noteDataSnapshotChild.getValue(ProdukPesanan.class);
18	
19	produk.setIdProdukPesanan(noteDataSnapshotChild.getKey());
20	listPesananBelumBayar.add(produk);
21	}
22	}
23	}
24	return listPesananBelumBayar;
25	}
26	}

Kode fungsi `getPesananBelumBayar()` pada Tabel 6.15 bertugas untuk mengambil data pesanan yang belum dibayar. Pertama melakukan perulangan pada dataSnapshot untuk menyimpan data pesanan ke object Pesanan. Lalu melakukan pengecekan kondisi jika status pesanan bernilai 1 dan username pemesanan sama dengan username pengguna maka dilakukan perulangan sekali lagi untuk mendapatkan data produk pesanan dari database lalu disimpan pada objek ProdukPesanan. Setelah itu data objek ProdukPesanan disimpan dalam arrayList.

**Tabel 6.16 Kode Program Fungsi getSaldo()**

1	public class M_User {
2	public int getSaldo(String username, DataSnapshot
3	dataSnapshot){
4	int saldo = 0;
5	for (final DataSnapshot noteDataSnapshot :
6	dataSnapshot.getChildren()){
7	Pembeli pembeli =
8	noteDataSnapshot.getValue(Pembeli.class);
9	
10	pembeli.setUsername(noteDataSnapshot.getKey());
11	if
12	(pembeli.getUsername().equalsIgnoreCase(username)){
13	saldo = pembeli.getSaldo();
14	}
15	}
16	return saldo;
17	}
18	}

Kode fungsi getSaldo() pada Tabel 6.16 bertugas untuk mengambil data saldo dari database. Pertama inialisasi saldo dengan nilai 0, lalu melakukan perulangan untuk mengambil nilai. Didalam perulangan tersebut terdapat pengecekan kondisi jika username sama dengan username pembeli maka akan dilakukan pengambiln data saldo sesuai username tersebut, tetelah itu melakukan pengembalian nilai saldo.

**Tabel 6.17 Kode Program Fungsi bayar()**

1	public class C_Transaksi {
2	public boolean bayar(int saldo, final String
3	kodePesanan, final int total, final String username){
4	mDatabase =
5	FirestoreDatabase.getInstance().getReference();
6	final int kembalian = saldo - total;
7	if (kembalian >= 0){
8	new M_User().potongSaldo(username, kembalian);
9	new
10	M_Pesanan().updateStatusPesanan(kodePesanan);
11	return true;
12	} else {
13	return false;
14	}
15	}
16	}

Kode fungsi bayar() pada Tabel 6.17 bertugas untuk melakukan pengecekan kondisi jika jumlah saldo lebih dari total harga maka akan memanggil fungsi potongSaldo() dan updateStatusPesanan(). Jika tidak maka mengembalikan nilai *false*.

**Tabel 6.18 Kode Program Fungsi potongSaldo()**

1	public class M_User {
2	public void potongSaldo(String username, int
3	kembalian){
4	mDatabase =
5	FirestoreDatabase.getInstance().getReference();

6	
7	<code>mDatabase.child("user").child(username).child("saldo").set</code>
8	<code>Value(kembalian);</code>
9	<code>}</code>
10	<code>}</code>

Kode fungsi `potongSaldo()` pada Tabel 6.18 bertugas untuk mengubah total saldo yang dimiliki menjadi saldo kembalian dari hasil pembayaran.

**Tabel 6.19 Kode Program Fungsi `updateStatusPesanan()`**

1	<code>public class M_Pesanan {</code>
2	<code>    public void updateStatusPesanan(String kodePesanan){</code>
3	<code>        mDatabase =</code>
4	<code>        FirebaseDatabase.getInstance().getReference();</code>
5	
6	<code>        mDatabase.child("pesanan").child(kodePesanan).child("statu</code>
7	<code>sPembayaran").setValue("1");</code>
8	<code>    }</code>
9	<code>}</code>

Kode fungsi `updateStatusPesanan()` pada Tabel 6.19 bertugas untuk mengubah status pembayaran menjadi telah dibayar. Pertama melakukan instansiasi `FirebaseDatabase`, lalu melakukan perubahan data pada `statusPembayaran` menjadi "1".

### 6.2.3.3 Kode Program Mengisi Saldo E-Dompet

Pada fungsional mengisi saldo E-dompet penjualan terdapat beberapa kode fungsi yang disematkan. Berikut tabel Kode Program yang dapat dilihat pada Tabel 6.20 sampai Tabel 6.25.

**Tabel 6.20 Kode Program Fungsi `generatedDepositPlus()`**

1	<code>public class C_Transaksi {</code>
2	<code>    public int generatedDepositPlus(int nominal,</code>
3	<code>    DataSnapshot dataSnapshot){</code>
4	<code>        boolean cek = false;</code>
5	<code>        int randomNum = 1 + (int)(Math.random() * 999);</code>
6	<code>        int depositPlus = nominal+randomNum;</code>
7	<code>        while (cek == false){</code>
8	<code>            for (DataSnapshot noteDataSnapshot :</code>
9	<code>dataSnapshot.getChildren()){</code>
10	<code>                Transaksi transaksi = new Transaksi();</code>
11	
12	<code>                transaksi.setUser(noteDataSnapshot.getKey());</code>
13	<code>                for (DataSnapshot tNoteDataSnapshot :</code>
14	<code>noteDataSnapshot.getChildren()){</code>
15	<code>                    transaksi =</code>
16	<code>tNoteDataSnapshot.getValue(Transaksi.class);</code>
17	
18	<code>                transaksi.setIdTransaksi(tNoteDataSnapshot.getKey());</code>
19	<code>                if (transaksi.getDepositPlus() ==</code>
20	<code>depositPlus &amp;&amp;</code>
21	<code>transaksi.getStatusDeposit().equalsIgnoreCase("0")){</code>
22	<code>                    randomNum = 1 +</code>
23	<code>(int)(Math.random() * 999);</code>
24	<code>                    depositPlus = nominal+randomNum;</code>
25	<code>                }</code>
26	<code>        }</code>

27	}
28	cek = true;
29	}
30	return depositPlus;
31	}
32	}
33	

Kode fungsi generatedDepositPlus() pada Tabel 6.20 bertugas untuk menginisialisasi depositPlus dengan format nominalDeposit ditambah dengan 3 digit angka acak. Terdapat pula pengecekan kondisi jika depositPlus telah ada sebelumnya maka akan dilakukan perulangan inisialisasi depositPlus.

**Tabel 6.21 Kode Program Fungsi insertTransaksiDeposit()**

1	public class M_Transaksi {
2	public void insertTransaksiDeposit(int depositPlus,
3	String username, int nominal, String idBank){
4	String id = "TID"+new
5	SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date());
6	mDatabase =
7	FirebaseDatabase.getInstance().getReference();
8	
9	mDatabase.child("transaksi").child(username).child(id).set
10	Value(new Transaksi(depositPlus, nominal, "0", idBank));
11	}
12	}

Kode fungsi insertTransaksiDeposit() pada Tabel 6.21 bertugas untuk menyimpan data transaksi deposit pada database firebase. Pertama melakukan inisialisasi idTransaksi dengan nilai "TID"+new SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date()), melakukan instansiasi FirebaseDatabase. Lalu melakukan penambahan data Transaksi.

**Tabel 6.22 Kode Program Fungsi getTransaksi()**

1	public class M_Transaksi {
2	public Transaksi getTransaksi(final int depositPlus,
3	DataSnapshot dataSnapshot){
4	Transaksi transaksi = new Transaksi();
5	for (DataSnapshot noteDataSnapshot :
6	dataSnapshot.getChildren()){
7	for (DataSnapshot childDataSnapshot :
8	noteDataSnapshot.getChildren()){
9	if (depositPlus ==
10	Integer.parseInt(childDataSnapshot.child("depositPlus").ge
11	tValue().toString())){
12	transaksi = new
13	Transaksi(childDataSnapshot.getKey(),
14	noteDataSnapshot.getKey(),
15	depositPlus,
16	
17	Integer.parseInt(childDataSnapshot.child("deposit").getVal
18	ue().toString()),
19	
20	childDataSnapshot.child("statusDeposit").getValue().toStri
21	ng(),
22	
23	childDataSnapshot.child("idBank").getValue().toString());

24	}
25	}
26	}
27	return transaksi;
28	}
29	}

Kode fungsi `getTransaksi()` pada Tabel 6.22 bertugas untuk mendapatkan data transaksi dari database firebase. Pertama melakukan instansiasi obyek Transaksi melakukan perulangan dan didalam perulangan tersebut dilakukan perulangan lagi untuk mencari nilai Transaksi. Pada perulangan terakhir terdapat pengecekan kondisi jika nilai `depositPlus` sama dengan nilai `depositPlus` pada database maka dilakukan pengambilan nilai data Transaksi lalu melakukan pengembalian nilai Transaksi tersebut.

**Tabel 6.23 Kode Program Fungsi notifBank()**

1	public class C_Transaksi {
2	public void notifBank(final int depositPlus){
3	mDatabase =
4	FirebaseDatabase.getInstance().getReference();
5	
6	mDatabase.child("transaksi").addListenerForSingleValueEven
7	t(new ValueEventListener() {
8	@Override
9	public void onDataChange(DataSnapshot
10	dataSnapshot) {
11	for (DataSnapshot noteDataSnapshot :
12	dataSnapshot.getChildren()){
13	for (DataSnapshot childDataSnapshot :
14	noteDataSnapshot.getChildren()){
15	if (depositPlus ==
16	Integer.parseInt(childDataSnapshot.child("depositPlus").ge
17	tValue().toString())){
18	final String username =
19	noteDataSnapshot.getKey();
20	String id =
21	childDataSnapshot.getKey();
22	final int deposit =
23	Integer.parseInt(childDataSnapshot.child("deposit").getVal
24	ue().toString());
25	new
26	M_Transaksi().ubahStatus(username, id, "1");
27	new
28	M_User().tambahSaldo(username, deposit);
29	}
30	}
31	}
32	}
33	
34	@Override
35	public void onCancelled(DatabaseError
36	databaseError) {
37	
38	}
39	});
40	}
41	}

42	
43	

Kode fungsi notifBank() pada Tabel 6.23 bertugas untuk mendapatkan data transaksi deposit sesuai dengan depositPlus yang dimasukkan. Jika data berhasil didapat maka selanjutnya memanggil fungsi ubahStatus() dan tambahSaldo().

**Tabel 6.24 Kode Program Fungsi ubahStatus()**

1	public class M_Transaksi {
2	public void ubahStatus(String username, String
3	idTransaksi, String statusTransaksi){
4	mDatabase =
5	FirebaseDatabase.getInstance().getReference();
6	
7	mDatabase.child("transaksi").child(username).child(idTrans
8	aksi).child("statusDeposit").setValue(statusTransaksi);
9	}
10	}

Kode fungsi ubahStatus() pada Tabel 6.24 bertugas untuk mengubah statusDeposit yang sebelumnya belum dibayar menjadi telah dibayar. Pertama melakukan instansiasi FirebaseDatabase, lalu melakukan perubahan data pada statusTransaksi sesuai dengan parameter.

**Tabel 6.25 Kode Program Fungsi tambahSaldo()**

1	public class M_User {
2	public void tambahSaldo(final String username, final
3	int deposit){
4	mDatabase =
5	FirebaseDatabase.getInstance().getReference();
6	
7	mDatabase.child("user").child(username).child("saldo").add
8	ListenerForSingleValueEvent(new ValueEventListener() {
9	@Override
10	public void onDataChange(DataSnapshot
11	dataSnapshot) {
12	int saldoAwal =
13	Integer.parseInt(dataSnapshot.getValue().toString());
14	
15	mDatabase.child("user").child(username).child("saldo").set
16	Value(saldoAwal+deposit);
17	}
18	
19	@Override
20	public void onCancelled(DatabaseError
21	databaseError) {
22	
23	}
24	});
25	}
26	}

Kode fungsi tambahSaldo() pada Tabel 6.25 bertugas untuk mengubah saldo awal menjadi saldo yang telah ditambahkan dengan nominal deposit. Pertama melakukan instansiasi FirebaseDatabase, lalu melakukan perubahan data pada saldo user.



#### 6.2.3.4 Kode Program Edit Status Pesanan

Pada fungsional mengisi saldo E-dompet penjualan terdapat beberapa kode fungsi yang disematkan. Berikut salah satu tabel Kode Program yang dapat dilihat pada Tabel 6.26

**Tabel 6.26 Kode Program Fungsi setStatusPesanan()**

1	public class M_Pesanan {
2	public void setStatusPesanan(final String idProduk,
3	final String status){
4	mDatabase =
5	FirebaseDatabase.getInstance().getReference();
6	
7	mDatabase.child("pesanan").addListenerForSingleValueEvent (
8	new ValueEventListener() {
9	@Override
10	public void onDataChange (DataSnapshot
11	dataSnapshot) {
12	for (DataSnapshot noteDataSnapshot :
13	dataSnapshot.getChildren()) {
14	Pesanan pesanan =
15	noteDataSnapshot.getValue (Pesanan.class);
16	
17	pesanan.setNomerPesanan (noteDataSnapshot.getKey());
18	for (DataSnapshot
19	noteDataSnapshotChild :
20	noteDataSnapshot.child("ProdukPesanan").getChildren()) {
21	ProdukPesanan produkPesanan =
22	noteDataSnapshotChild.getValue (ProdukPesanan.class);
23	
24	produkPesanan.setIdProdukPesanan (noteDataSnapshotChild.get
25	Key());
26	
27	if (produkPesanan.getIdProdukPesanan().equalsIgnoreCase (idP
28	roduk)) {
29	mDatabase =
30	FirebaseDatabase.getInstance().getReference();
31	
32	mDatabase.child("pesanan").child(pesanan.getNomerPesanan()
33	).child("ProdukPesanan").child(idProduk).child("statusProd
34	ukPesanan").setValue (status);
35	}
36	}
37	}
38	}
39	
40	@Override
41	public void onCancelled (DatabaseError
42	databaseError) {
43	
44	}
45	});
46	}
47	}
48	

Kode fungsi setStatusPesanan() pada Tabel 6.26 bertugas untuk mengubah status produk pesanan sesuai dengan parameter status dan idProduk. Sebelum

melakukan perubahan status, fungsi ini melakukan pengambilan data menurut idProdukPesanan untuk mendapatkan data yang sesuai.

### 6.2.3.5 Kode Program Update Produk

Pada fungsional update produk terdapat beberapa kode fungsi yang disematkan. Berikut tabel Kode Program yang dapat dilihat pada Tabel 6.27 sampai Tabel 6.28.

**Tabel 6.27 Kode Program Fungsi tampilDialogFormUP()**

1	public class V_KelolaProduk {
2	public void tampilDialogFormUP(View v, final Produk
3	produk) {
4	AlertDialog.Builder dialogBuilder = new
5	AlertDialog.Builder(v.getContext());
6	LayoutInflater inflater =
7	LayoutInflater.from(v.getContext());
8	final View dialogView =
9	inflater.inflate(R.layout.edit_produk, null);
10	dialogBuilder.setView(dialogView);
11	
12	final EditText editTextNama = (EditText)
13	dialogView.findViewById(R.id.editTextNama);
14	editTextNama.setText(produk.getNamaProduk());
15	final EditText editTextHarga = (EditText)
16	dialogView.findViewById(R.id.editTextHarga);
17	
18	editTextHarga.setText(Integer.toString(produk.getHarga()))
19	;
20	final EditText editTextJumlah = (EditText)
21	dialogView.findViewById(R.id.editTextJumlah);
22	
23	editTextJumlah.setText(Integer.toString(produk.getStok()))
24	;
25	final Button buttonEdit = (Button)
26	dialogView.findViewById(R.id.buttonEditProduk);
27	final Button buttonCancel = (Button)
28	dialogView.findViewById(R.id.buttonCancel);
29	
30	dialogBuilder.setTitle("Edit Produk");
31	final AlertDialog b = dialogBuilder.create();
32	b.show();
33	
34	buttonEdit.setOnClickListener(new
35	View.OnClickListener() {
36	@Override
37	public void onClick(View view) {
38	String name =
39	editTextNama.getText().toString().trim();
40	String harga =
41	editTextHarga.getText().toString();
42	String jumlah =
43	editTextJumlah.getText().toString();
44	if (!TextUtils.isEmpty(name) &&
45	!TextUtils.isEmpty(harga) && !TextUtils.isEmpty(jumlah)) {
46	SharedPreferences pref =
47	view.getContext().getSharedPreferences("MyPref", 0);

48	new
49	C_KelolaProduk().updateProduk(produk.getIdProduk(),
50	pref.getString("username", null), name,
51	Integer.parseInt(harga), Integer.parseInt(jumlah));
52	b.dismiss();
53	}
54	}
55	});
56	
57	buttonCancel.setOnClickListener(new
58	View.OnClickListener() {
59	@Override
60	public void onClick(View view) {
61	b.dismiss();
62	}
63	});
64	}
65	}
66	

Kode fungsi tampilDialogFormUP() pada Tabel 6.27 bertugas untuk menampilkan dialog untuk Form Update Produk. Pertama instansiasi AlertDialog dan mengatur isi AlertDialog dan form-form yang akan ditampilkan lalu melakukan penampilan AlertDialog. Pada AlertDialog jika menekan tombol edit maka akan mengambil semua nilai form yang diinputkan dan memasukkan pada fungsi updateProduk(). Jika yang ditekan cencel maka akan menutup AlertDialog.

**Tabel 6.28 Kode Program Fungsi updateProduk()**

1	public class M_Produk {
2	public void updateProduk(String username, String id,
3	int stok){
4	mDatabase =
5	FirestoreDatabase.getInstance().getReference();
6	
7	mDatabase.child("user").child(username).child("produk").ch
8	ild(id).child("namaProduk").setValue(nama);
9	
10	mDatabase.child("user").child(username).child("produk").ch
11	ild(id).child("harga").setValue(harga);
12	
13	mDatabase.child("user").child(username).child("produk").ch
14	ild(id).child("stok").setValue(stok);
15	}
16	}

Kode fungsi updateProduk() pada Tabel 6.28 bertugas untuk melakukan perubahan data pada Produk. Pertama melakukan instansiasi FirestoreDatabase, lalu melakukan perubahan data pada nama Produk, harga Produk, dan stok Produk.

#### 6.2.3.6 Kode Program Tambah Praduk

Pada fungsional tambah produk terdapat beberapa kode fungsi yang disematkan. Berikut salah satu tabel Kode Program yang dapat dilihat pada Tabel 6.29.

**Tabel 6.29 Kode Program Fungsi insertProduk()**

1	public class M_User {
2	public void insertProduk(String username, String nama,
3	int jumlah, int harga, String tipe, String foto){
4	String id = "PID"+new
5	SimpleDateFormat("yyyyMMddHHmmssSS").format(new Date());
6	mDatabase =
7	FirebaseDatabase.getInstance().getReference();
8	if (foto.equalsIgnoreCase("images/null")){
9	
10	mDatabase.child("user").child(username).child("produk").ch
11	ild(id).setValue(new Produk(nama, jumlah, harga, tipe));
12	} else {
13	
14	mDatabase.child("user").child(username).child("produk").ch
15	ild(id).setValue(new Produk(nama, jumlah, harga, tipe,
16	foto));
17	}
18	}
19	}
20	

Kode fungsi insertProduk() pada Tabel 6.29 bertugas untuk melakukan penambahan data pada Produk. Pertama melakukan inisialisasi id dan instansiasi FirebaseDatabase, lalu melakukan pengecekan kondisi jika tidak ada foto produk maka dilakukan penambahan Produk tanpa foto, selain itu maka akan dilakukan penambahan Produk dengan foto Produk.

#### 6.2.3.7 Kode Program Hapus Produk

Pada fungsional hapus produk terdapat beberapa kode fungsi yang disematkan. Berikut salah satu tabel Kode Program yang dapat dilihat pada Tabel 6.30.

**Tabel 6.30 Kode Program Fungsi hapusProduk()**

1	public class M_Produk {
2	public void hapusProduk(String ID, String username){
3	mDatabase =
4	FirebaseDatabase.getInstance().getReference();
5	
6	mDatabase.child("user").child(username).child("produk").ch
7	ild(ID).removeValue();
8	}
9	}
10	

Kode fungsi hapusProduk() pada Tabel 6.30 bertugas untuk melakukan penghapusan data pada Produk. Pertama melakukan instansiasi FirebaseDatabase, lalu melakukan penghapusan data pada Produk.

#### 6.2.3.8 Kode Program Edit Status Toko (iterasi ke-1 fase konstruksi)

Pada fungsional Edit Status Toko terdapat beberapa kode fungsi yang disematkan. Berikut salah satu tabel Kode Program yang dapat dilihat pada Tabel 6.31.

**Tabel 6.31 Kode Program Fungsi editToko()**

1	public class M_User {
2	public void editToko(boolean isChecked, View v){
3	SharedPreferences pref =
4	v.getContext().getSharedPreferences("MyPref", 0);
5	final SharedPreferences.Editor editor =
6	pref.edit();
7	if (isChecked){
8	
9	mDatabase.child("user").child(pref.getString("username",
10	null)).child("statusToko").setValue("1");
11	editor.putString("statusToko", "1");
12	editor.commit();
13	} else {
14	
15	mDatabase.child("user").child(pref.getString("username",
16	null)).child("statusToko").setValue("0");
17	editor.putString("statusToko", "0");
18	editor.commit();
19	}
20	}
21	}

Kode fungsi Tabel 6.31 untuk melakukan edit toko penjual apakah sedang buka atau sedang tutup. Pertama melakukan instansiasi SharedPreferences lalu instansiasi SharedPreferences.Editor. Setelah itu terdapat pengecekan kondisi apakah tombol *Switch* telah aktif atau belum, jika aktif maka mengubah status menjadi buka, jika tidak maka status dirubah menjadi tutup.

#### 6.2.3.9 Kode Program Edit Status Pesanan Diambil (iterasi ke-1 fase konstruksi)

Pada fungsional Edit Status Pesanan Diambil terdapat beberapa kode fungsi yang disematkan. Berikut salah satu tabel Kode Program yang dapat dilihat pada Tabel 6.32.

**Tabel 6.32 Kode Program Fungsi tampilDialogPengambilan()**

1	public class V_Pesanan {
2	public void tampilDialogPengambilan(View v, String
3	produk, String username, int jumlah, final String
4	IdProdukPesanan){
5	AlertDialog.Builder dialogBuilder = new
6	AlertDialog.Builder(v.getContext());
7	LayoutInflater inflater =
8	LayoutInflater.from(v.getContext());
9	final View dialogView =
10	inflater.inflate(R.layout.popup_ambil, null);
11	dialogBuilder.setView(dialogView);
12	
13	final TextView produk_pesanan = (TextView)
14	dialogView.findViewById(R.id.produk_pesanan);
15	produk_pesanan.setText(produk);
16	final TextView pemesan = (TextView)
17	dialogView.findViewById(R.id.pemesan);
18	pemesan.setText(username);
19	final TextView jumlah_pesan = (TextView)
20	dialogView.findViewById(R.id.jumlah_pesan);
21	

```

22         jumlah_pesanan.setText("Qty:
23 "+Integer.toString(jumlah));
24         final Button buttonAmbil = (Button)
25         dialogView.findViewById(R.id.buttonAmbil);
26
27         dialogBuilder.setTitle("Pengambilan Pesanan");
28         final AlertDialog b = dialogBuilder.create();
29         b.show();
30
31         buttonAmbil.setOnClickListener(new
32         View.OnClickListener() {
33             @Override
34             public void onClick(View v) {
35                 new
36                 C_Pesanan().setStatusPesanan(IdProdukPesanan,"3");
37                 b.dismiss();
38             }
39         });
40     }

```

Kode fungsi Tabel 6.32 untuk menampilkan dialog detail produk pesanan agar dapat diperlihatkan ke penjual. Pertama instansiasi AlertDialog dan mengatur isi AlertDialog dan beberapa *TextView* yang akan ditampilkan lalu melakukan penampilan AlertDialog. Pada AlertDialog jika menekan tombol Ambil maka akan memanggil fungsi setStatusPesanan() dengan parameter idPesanan dan status produk pesanan bernilai "3" yang artinya status telah diambil.

#### 6.2.4 Implementasi Antarmuka Pengguna (UI)

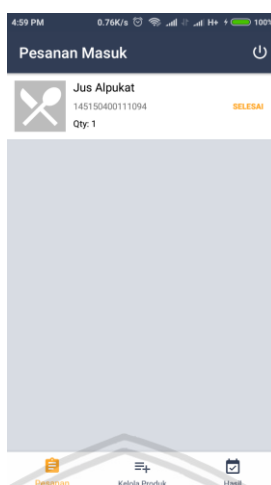
Tahap implementasi antarmuka pengguna ini merupakan proses implementasi tampilan sistem yang telah dibangun agar pengguna dapat berinteraksi dengan sistem informasi pemesanan dan penjualan pada kantin filkom. Antarmuka pada sistem pemesanan dan penjualan ini terdapat 2 pengguna yaitu Antarmuka Pembeli dan Antarmuka Penjual.

##### Antarmuka Penjual

Antarmuka pada pembeli merupakan antarmuka yang digunakan oleh seorang penjual melakukan penjualan di kantin filkom. Berikut adalah implementasi antarmuka untuk tampilan penjual yang ada pada sistem yang dapat dilihat pada Gambar 6.5 sampai Gambar 6.12.



## 1. Implementasi Antarmuka Pesanan



**Gambar 6.4 Implementasi Antarmuka Pesanan**

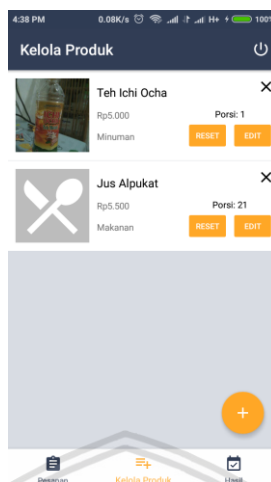
Antarmuka pada Gambar 6.5 ini untuk melihat list pesanan yang dipesan oleh pembeli ke penjual. Antarmuka ini juga dapat edit status pesanan pembeli menjadi selesai. Pada antarmuka ini terdapat perubahan tampilan dan penambahan fungsional untuk melakukan pembatalan pesanan pada iterasi ke-1 fase konstruksi. Perubahan tersebut dapat dilihat pada Gambar 6.6.



**Gambar 6.5 Implementasi Antarmuka Pesanan (iterasi ke-1 fase konstruksi)**

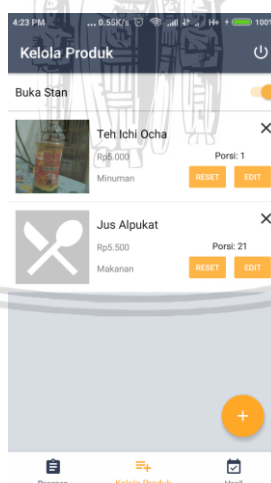
Pada Gambar 6.6 terdapat perubahan tampilan yang terdapat penambahan tombol “x” untuk melakukan pembatalan pesanan jika penjual tidak ingin melayani atau stok telah habis. Terdapat pula penambahan tampilan agar penjual juga dapat melihat daftar pesanan yang belum diambil.

## 2. Implementasi Antarmuka Kelola Produk



**Gambar 6.6 Implementasi Antarmuka Kelola Produk**

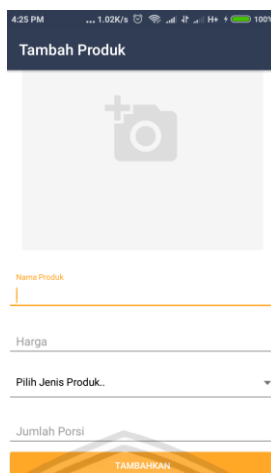
Antarmuka pada Gambar 6.7 ini untuk menampilkan produk penjualan penjual dan melakukan pengelolaan produk penjualan. Pada antarmuka ini dapat melakukan edit, tambah, hapus produk dan reset stok produk. Pada antarmuka ini terdapat perubahan tampilan dan penambahan fungsional untuk melakukan edit status toko pada iterasi ke-1 fase konstruksi agar ketika stan penjual sudah tutup pembeli tidak dapat memesan lagi. Perubahan tersebut dapat dilihat pada Gambar 6.8.



**Gambar 6.7 Implementasi Antarmuka Kelola Produk (iterasi ke-1 fase konstruksi)**

Gambar 6.8 terdapat perubahan tampilan yaitu penambahan tombol saklar/*swich* untuk melakukan edit status toko. Ketika tombol saklar hidup artinya status toko sedang buka, jika tombol saklar dalam keadaan mati maka status toko sedang tutup.

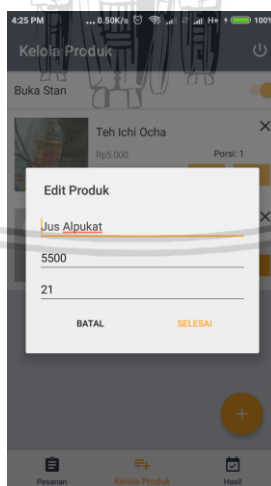
### 3. Implementasi Antarmuka Tambah Produk



**Gambar 6.8 Implementasi Antarmuka Tambah Produk**

Antarmuka pada Gambar 6.9 ini akan tampil ketika penjual menekan tombol “+” pada antarmuka Kelola Produk. Antarmuka ini digunakan untuk melakukan pengisian produk yang akan ditambah. Ketika tombol “+” pada kelola produk ditekan akan tampil antarmuka seperti pada Gambar 6.9, lalu ketika menekan gambar kamera akan berpindah ke aplikasi kamera untuk menangkap visual dari produk yang ingin dijual. Pada inputan terdapat nama produk, harga, jenis dan jumlah, dan terdapat tombol tambahkan yang jika ditekan akan menambah produk.

### 4. Implementasi Antarmuka Update Produk

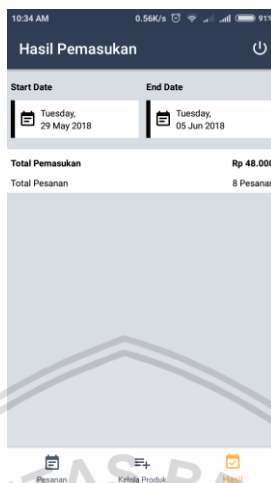


**Gambar 6.9 Implementasi Antarmuka Update Produk**

Antarmuka pada Gambar 6.10 ini akan tampil ketika penjual menekan tombol “edit” pada antarmuka Kelola Produk. Antarmuka ini untuk melakukan update produk penjualan. Terdapat inputan nama produk harga dan jumlah, lalu ketika tombol selesai ditekan maka akan mengubah produk sesuai inputan yang diisi,

namun ketika menekan tombol batal sistem tidak akan mengubah info produk meskipun telah dirubah inputan sebelumnya.

## 5. Implementasi Antarmuka Hasil Penjualan



**Gambar 6.10 Implementasi Antarmuka Hasil Penjualan**

Antarmuka pada Gambar 6.11 ini untuk menampilkan hasil penjualan yang telah dilakukan penjual pada hari tertentu. Awal tampilan pada antarmuka ini akan menampilkan hasil penjualan pada hari ini, lalu ketika mengatur *range* tanggal yang diinginkan akan menampilkan hasil penjualan pada *range* tanggal yang telah disesuaikan.



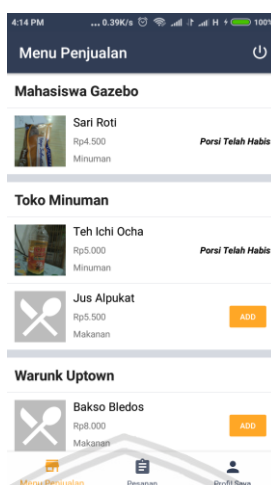
**Gambar 6.11 Implementasi Antarmuka Atur Rentang Tanggal Hasil Penjualan**

Antarmuka pada Gambar 6.12 ini digunakan untuk memilih tanggal yang diinginkan untuk mengatur rentang tanggal hasil penjualan.

### Antarmuka Pembeli

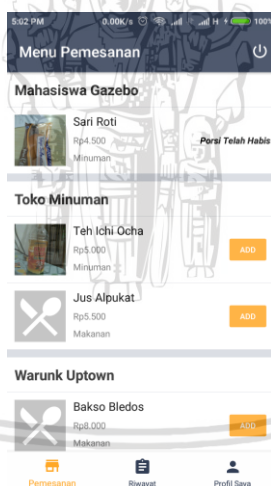
Antarmuka pada pembeli merupakan antarmuka yang digunakan oleh seorang pembeli yang ingin memesan makanan atau minuman pada kantin filkom. Berikut adalah implementasi antarmuka untuk tampilan pembeli yang ada pada sistem yang dapat dilihat pada Gambar 6.13 sampai Gambar 6.21.

## 1. Implementasi Antarmuka Menu Penjualan



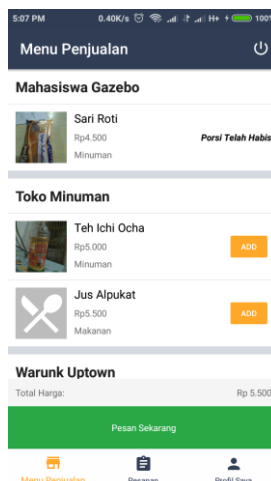
**Gambar 6.12 Implementasi Antarmuka Menu Penjualan**

Antarmuka pada Gambar 6.13 ini digunakan untuk melakukan pemesanan pada produk yang tersedia pada kantin filkom. Pada antarmuka ini dapat melakukan edit, tambah, hapus produk dan reset stok produk. Pada antarmuka ini terdapat perubahan tampilan pada iterasi ke-2 fase konstruksi yaitu pada penulisan menu navigasi karena terdapat kata ambigu pada menu navigasi.



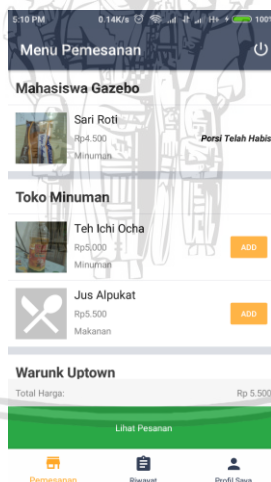
**Gambar 6.13 Implementasi Antarmuka Menu Penjualan (iterasi ke-2 fase konstruksi)**

Gambar 6.14 terdapat perubahan kata pada menu navigasi dan judul tampilan. Judul tampilan berubah menjadi “Menu Pemesanan” dan pada navigasi menjadi “Pemesanan”.



**Gambar 6.14 Implementasi Antarmuka Menu Penjualan Telah Memesan**

Antarmuka pada Gambar 6.15 ini akan tampil ketika pembeli menekan tombol “ADD” dan sistem akan melakukan kalkulasi total harga yang telah dipesan. Untuk melihat rincian pembayaran pembeli harus menekan tombol “Pesan Sekarang”. Pada antarmuka ini juga terdapat perubahan tampilan pada iterasi ke-2 fase konstruksi yaitu penulisan pada tombol “Pesan Sekarang” karena terdapat kata ambigu pada tombol tersebut.

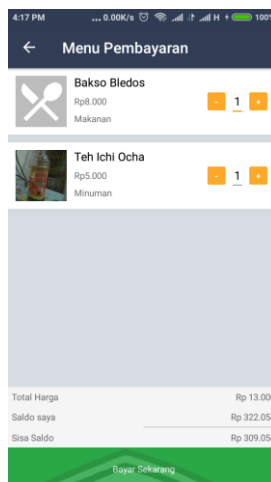


**Gambar 6.15 Implementasi Antarmuka Menu Penjualan Telah Memesan (iterasi ke-2 fase konstruksi)**

Gambar 6.16 terdapat perubahan penulisan pada tombol yang sebelumnya “Pesan Sekarang” menjadi “Lihat Pesanan” serta pada navigasi yang sebelumnya “Menu Penjualan” menjadi “Pesanan”.



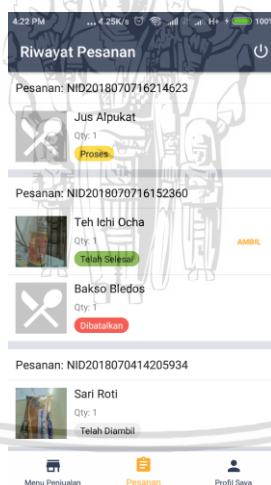
## 2. Implementasi Antarmuka Pembayaran



**Gambar 6.16 Implementasi Antarmuka Pembayaran**

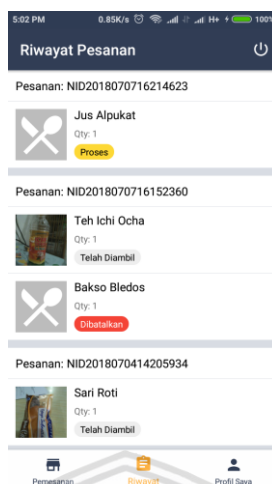
Antarmuka pada Gambar 6.17 ini untuk melihat detail list pesanan yang telah dipesan pembeli dan detail pembayaran. Antarmuka ini dapat melakukan penambahan jumlah pesanan ataupun pengurangan jumlah pesanan dan pada bagian bawah terdapat informasi total harga, saldo, serta karkulasi sisa saldo.

## 3. Implementasi Antarmuka Riwayat Pemesanan



**Gambar 6.17 Implementasi Antarmuka Riwayat Pemesanan**

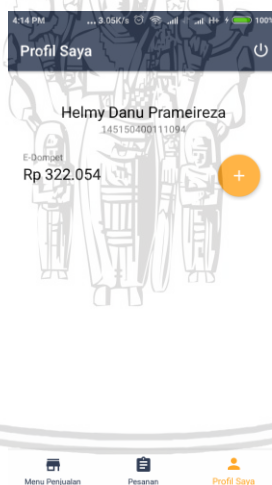
Antarmuka pada Gambar 6.18 ini untuk melihat riwayat pesanan pembeli yang telah dibayar. Terdapat pula informasi status pesanan yang telah dipesan yaitu proses, telah selesai, telah diambil dan dibatalkan.



**Gambar 6.18 Implementasi Antarmuka Riwayat Pemesanan (iterasi ke-2 fase konstruksi)**

Gambar 6.19 terdapat perubahan di penulisan menu navigasi tampilan pada iterasi ke-3 fase konstruksi yaitu penulisan yang sebelumnya “Pesanan” menjadi “Riwayat”.

#### 4. Implementasi Antarmuka Profil Akun



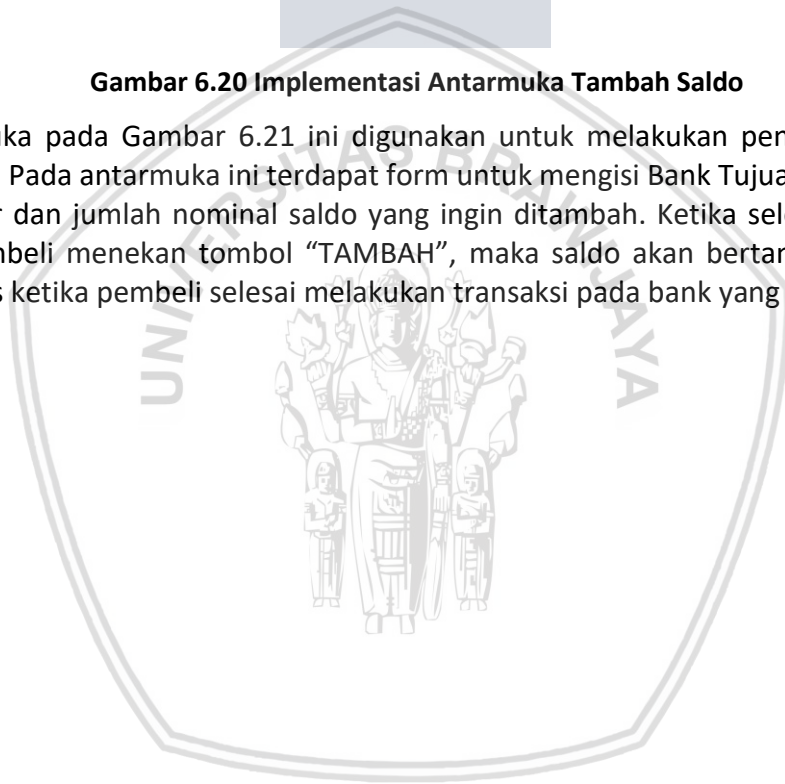
**Gambar 6.19 Implementasi Antarmuka Profil Akun**

Antarmuka pada Gambar 6.20 ini untuk melihat informasi akun profil pembeli berupa nama, username, dan informasi saldo pembeli. Pada antarmuka ini juga terdapat tombol tambah saldo yang berfungsi untuk menampilkan antarmuka Tambah Saldo.

## 5. Implementasi Antarmuka Tambah Saldo

**Gambar 6.20 Implementasi Antarmuka Tambah Saldo**

Antarmuka pada Gambar 6.21 ini digunakan untuk melakukan pengisian saldo pembeli. Pada antarmuka ini terdapat form untuk mengisi Bank Tujuan yang ingin ditranfer dan jumlah nominal saldo yang ingin ditambah. Ketika selesai mengisi dan pembeli menekan tombol “TAMBAH”, maka saldo akan bertambah secara otomatis ketika pembeli selesai melakukan transaksi pada bank yang dituju.



## BAB 7 PENGUJIAN SISTEM

Pengujian merupakan bagian dari tahap Relation Unified Proses yaitu termasuk dalam fase konstruksi dan fase transisi. Fase konstruksi meliputi pengujian *black box* dan pengujian *Compatibility*, untuk fase transisi terdapat UAT (*User Acceptance Testing*) dan Pengujian Waktu. Tujuan dari bab pengujian pada fase konstruksi adalah untuk mengetahui apakah implementasi sistem terdapat cacat atau error yang terjadi. Pengujian pada fase transisi bertujuan untuk mengetahui apakah sistem yang telah dikembangkan sudah sesuai dan mampu menyelesaikan permasalahan yang ada dan untuk mengetahui penerimaan sistem oleh pengguna akhir.

### 7.1 Pengujian Black Box

Metode pengujian black-box fokus pengujian pada fungsional dari sistem. Selain itu pengujian dilakukan untuk mengetahui seberapa banyak kesalahan fungsi secara garis besar, antarmuka, akses database, inisialisasi dan tujuan akhir. Berikut hasil dari pengujian black-box.

#### 7.1.1 Rancangan Pengujian Black Box

Rancangan pengujian black-box adalah pendefinisian dari kondisi dan hasil yang diharapkan dari sistem. Rancangan berfungsi sebagai acuan dalam proses pengujian dan sebagai acuan penilaian apakah hasil dari pengujian tersebut berhasil atau gagal. Berikut rancangan pengujian black-box yang dapat dilihat pada Tabel 7.1.

Tabel 7.1 Perancangan Pengujian Black-Box

No.	Kode SRS	Pengujian	Test case	Harapan
1	SRF-01	Terdapat pesanan yang sedang proses dan telah selesai	Terdapat data produk pesanan dengan status proses dan telah selesai	Sistem menampilkan list pesanan yang belum selesai/proses dan yang belum diambil/telah selesai
		Tidak ada pesanan yang sedang proses, terdapat pesanan yang telah selesai	Tidak ada data produk pesanan dengan status proses, terdapat data produk pesanan dengan status telah selesai	Sistem hanya menampilkan list pesanan yang belum diambil

No.	Kode SRS	Pengujian	Test case	Harapan
		Tidak ada pesanan yang telah selesai, terdapat pesanan yang sedang proses	Tidak ada data produk pesanan dengan status telah selesai, terdapat data produk pesanan dengan status proses	Sistem hanya menampilkan list pesanan yang belum selesai/proses
		Tidak ada pesanan yang belum selesai/proses dan yang belum di ambil/telah selesai	Tidak ada data produk pesanan dengan status proses dan telah selesai	Sistem tidak menampilkan list pesanan yang belum selesai/proses maupun yang belum di ambil
2	SRF-02	Mengubah status pesanan menjadi telah selesai	Menekan tombol "Selesai"	Sistem mengubah status menjadi telah selesai dan menampilkannya pada list pesanan belum diambil
		Mengubah status pesanan menjadi dibatalkan	Menekan tombol "X"	Sistem mengubah status menjadi dibatalkan dan saldo pembeli dikembalikan
3	SRF-03	Berhasil menambah Produk dengan foto produk	Inputan diisi semua dan menekan tombol tambah	Sistem berhasil menambahkan produk baru pada list kelola produk dan menampilkan foto produknya
		Berhasil menambah Produk tidak dengan foto	Inputan diisi semua dan menekan tombol tambah kecuali	Sistem berhasil menambahkan produk baru pada list kelola produk dan menampilkan

No.	Kode SRS	Pengujian	Test case	Harapan
		produk	bagian upload foto	foto bawaan aplikasi
		Gagal menambahkan Produk	Terdapat inputan yang kosong selain upload foto	Sistem memberi peringatan jika harus mengisi semua inputan
4	SRF-04	Mengubah nama produk	Hanya mengganti nama produk pada inputan nama produk lalu tekan "selesai"	Sistem mengubah nama produk
		Mengubah harga produk	Hanya mengganti harga produk pada inputan harga produk lalu tekan "selesai"	Sistem mengubah harga produk
		Mengubah stok produk	Hanya mengganti stok produk pada inputan stok produk lalu tekan "selesai"	Sistem mengubah stok produk
		Mengubah nama dan harga produk	Mengganti nama dan harga produk pada inputan nama dan harga produk lalu tekan "selesai"	Sistem mengubah nama dan harga produk
		Mengubah nama dan stok produk	Mengganti nama dan stok produk pada inputan nama dan stok produk lalu tekan "selesai"	Sistem mengubah nama dan stok produk
		Mengubah harga	Mengganti harga	Sistem mengubah



No.	Kode SRS	Pengujian	Test case	Harapan
		dan stok produk	dan stok produk pada inputan harga dan stok produk lalu tekan "selesai"	harga dan stok produk
		Mengubah nama, harga dan stok produk	Mengganti nama, harga dan stok produk pada inputan nama, harga dan stok produk lalu tekan "selesai"	Sistem mengubah nama, harga dan stok produk
		Tidak mengubah produk	Tidak mengubah informasi pada inputan lalu tekan "selesai"	Sistem tidak mengubah informasi produk
		Tidak jadi mengubah produk	Menekan tombol "batal"	Sistem tidak mengubah informasi produk
5	SRF-05	Berhasil menghapus produk	Tekan tombol "X" lalu tekan Hapus	Sistem berhasil menghapus produk
		Batal menghapus produk	Tekan tombol "X" lalu tekan Batal	Sistem tidak menghapus produk
6	SRF-11	manampilkan total hasil pemasukan dan total pemesanan pada hari ini	Start Date dan End Date merujuk pada tanggal hari ini	Sistem manampilkan total hasil pemasukan dan total pemesanan pada hari ini

No.	Kode SRS	Pengujian	Test case	Harapan
		manampilkan total hasil pemasukan dan total pemesanan pada tanggal tertentu	Mengganti Start Date dan End Date agar merujuk pada tanggal tertentu	Sistem menampilkan total hasil pemasukan dan total pemesanan pada tanggal tertentu
		manampilkan total hasil pemasukan dan total pemesanan pada rentang tanggal tertentu	Mengganti Start Date dan End Date dengan rentang tanggal yang ditentukan	Sistem menampilkan total hasil pemasukan dan total pemesanan pada rentang tanggal tertentu
7	SRF-06	Terdapat produk yang dijual	Terdapat data produk penjualan dan status toko pada penjual sedang buka	Sistem menampilkan list produk yang dijual penjual
		Tidak ada produk penjualan	Tidak ada data produk penjualan atau semua status toko pada penjual sedang tutup	Sistem tidak menampilkan list produk yang dijual penjual
8	SRF-07	Berhasil menambah pesanan	Menekan tombol "add"	Sistem berhasil menambah pesanan, mengurangi stok produk dan menampilkan total harga beserta tombol "Pesan Sekarang"
		Menambah pesanan pada produk yang sama	Menekan tombol "add" pada produk yang sama atau menekan tombol "+" pada	Sistem berhasil ngubah jumlah pesanan dan total harga, serta mengurangi stok

No.	Kode SRS	Pengujian	Test case	Harapan
			menu pembayaran	produk
		Mengurangi produk pesanan	Menekan tombol "-" pada menu pembayaran	Sistem berhasil ngubah jumlah pesanan dan total harga, serta menambah stok produk
		Menghapus produk pesanan	Menekan tombol "-" pada menu pembayaran ketika jumlah produk pesanan 1	Sistem berhasil menghapus produk pesanan, serta menambah stok produk
9	SRF-08	Berhasil membayar pesanan	Jumlah total harga pesanan $\leq$ saldo yang dimiliki	Sistem berhasil mengubah status pesanan menjadi telah dibayar
		Gagal membayar pesanan	Jumlah total harga pesanan $>$ saldo yang dimiliki	Sistem memberi peringatan jika saldo tidak cukup
10	SRF-09	Berhasil mengisi saldo	Mengisi inputan bank tujuan tranfer dan nominal saldo yang akan diisi	Sistem menampilkan detail transaksi dan berhasil menambahkan saldo sesuai nominal yang diinput
		Gagal mengisi saldo	Terdapat inpuan yang kosong/tidak diisi	Sistem memberi peringatan harus mengisi semua inputan
11	SRF-12	Menampilkan data profil pengguna	Login menggunakan	Sistem berhasil mendapatkan dan

No.	Kode SRS	Pengujian	Test case	Harapan
			akun pembeli	menampilkan informasi profil pengguna dari season
12	SRF-10	Login berhasil menggunakan username penjual	Username penjual dan password benar	Sistem masuk ke tampilan penjual
		Login berhasil menggunakan username pembeli	Username pembeli dan password benar	Sistem masuk ke tampilan pembeli
		Username dan password kosong	Terdapat inpuan yang kosong/tidak diisi	Sistem memberi peringatan inputan harus diisi
		Username dan password salah	Username dan password salah	Sistem memberi peringatan username dan password salah
13	SRF-13	Registrasi berhasil	Semua inputan diisi dan username yang dimasukkan belum terdaftar	Sistem berhasil membuat user pembeli baru
		Gagal karena username telah ada	Menginputkan username yang telah terdaftar	Sistem menampilkan dialog peringatan username yang diinputkan sudah terdaftar sebelumnya
		Gagal karena email salah	Menginputkan email yang tidak sesuai format	Sistem memberi peringatan email salah
		Gagal karena inputan kosong	Terdapat inpuan yang kosong/tidak	Sistem memberi peringatan

No.	Kode SRS	Pengujian	Test case	Harapan
			diisi	inputan harus diisi
14	SRF-14	Mematikan tombol <i>switch</i>	Menekan tombol <i>switch</i> ketika sedang menyala atau seret tombol <i>switch</i> ke kiri.	Status toko akan berubah menjadi tutup
		Menghidupkan tombol <i>switch</i>	Menekan tombol <i>switch</i> ketika sedang mati atau seret tombol <i>switch</i> ke kanan.	Status toko akan berubah menjadi buka
15	SRF-15	Membatalkan pesanan	Menekan tombol "x" lalu pada dialog tekan tombol "ya"	Status pesanan berubah menjadi dibatalkan
		Tidak jadi membatalkan pesanan	Menekan tombol "x" lalu pada dialog tekan tombol "tidak"	Status pesanan tidak berubah.
16	SRF-16	Berhasil mengambil pesanan	Menekan tombol "lihat" lalu pada dialog tekan tombol "ambil"	Status pesanan berubah menjadi diambil.

### 7.1.2 Hasil Pengujian *Black Box*

Analisa dari hasil penerapan perancangan pengujian black-box. Teknik analisa yaitu membandingkan harapan yang telah didefinisikan pada perancangan dengan hasil yang ditampilkan oleh sistem. Berikut hasil dari pengujian black-box yang dapat dilihat pada Tabel 7.2.

**Tabel 7.2 Hasil Pengujian *Black Box***

No.	Pengujian	Harapan	Hasil
1	Terdapat pesanan yang sedang proses dan telah selesai	Sistem menampilkan list pesanan yang belum selesai/proses dan yang belum di ambil/telah selesai	<i>Valid</i>
2	Tidak ada pesanan yang sedang proses, terdapat pesanan yang telah selesai	Sistem hanya menampilkan list pesanan yang belum diambil	<i>Valid</i>
3	Tidak ada pesanan yang telah selesai, terdapat pesanan yang sedang proses	Sistem hanya menampilkan list pesanan yang belum selesai/proses	<i>Valid</i>
4	Tidak ada pesanan yang belum selesai/proses dan yang belum di ambil/telah selesai	Sistem tidak menampilkan list pesanan yang belum selesai/proses maupun yang belum di ambil	<i>Valid</i>
5	Mengubah status pesanan menjadi telah selesai	Sistem mengubah status menjadi telah selesai dan menampilkannya pada list pesanan belum diambil	<i>Valid</i>
6	Mengubah status pesanan menjadi dibatalkan	Sistem mengubah status menjadi dibatalkan dan saldo pembeli dikembalikan	<i>Valid</i>
7	Berhasil menambah Produk dengan foto produk	Sistem berhasil menambahkan produk baru pada list kelola produk dan menampilkan	<i>Valid</i>



No.	Pengujian	Harapan	Hasil
		foto produknya	
8	Berhasil menambah Produk tidak dengan foto produk	Sistem berhasil menambahkan produk baru pada list kelola produk dan menampilkan foto bawaan aplikasi	<i>Valid</i>
9	Gagal menambahkan Produk	Sistem memberi peringatan jika harus mengisi semua inputan	<i>Valid</i>
10	Mengubah nama produk	Sistem mengubah nama produk	<i>Valid</i>
11	Mengubah harga produk	Sistem mengubah harga produk	<i>Valid</i>
12	Mengubah stok produk	Sistem mengubah stok produk	<i>Valid</i>
13	Mengubah nama dan harga produk	Sistem mengubah nama dan harga produk	<i>Valid</i>
14	Mengubah nama dan stok produk	Sistem mengubah nama dan stok produk	<i>Valid</i>
15	Mengubah harga dan stok produk	Sistem mengubah harga dan stok produk	<i>Valid</i>
16	Mengubah nama, harga dan stok produk	Sistem mengubah nama, harga dan stok produk	<i>Valid</i>
17	Tidak mengubah produk	Sistem tidak mengubah informasi produk	<i>Valid</i>
18	Tidak jadi mengubah produk	Sistem tidak mengubah informasi produk	<i>Valid</i>

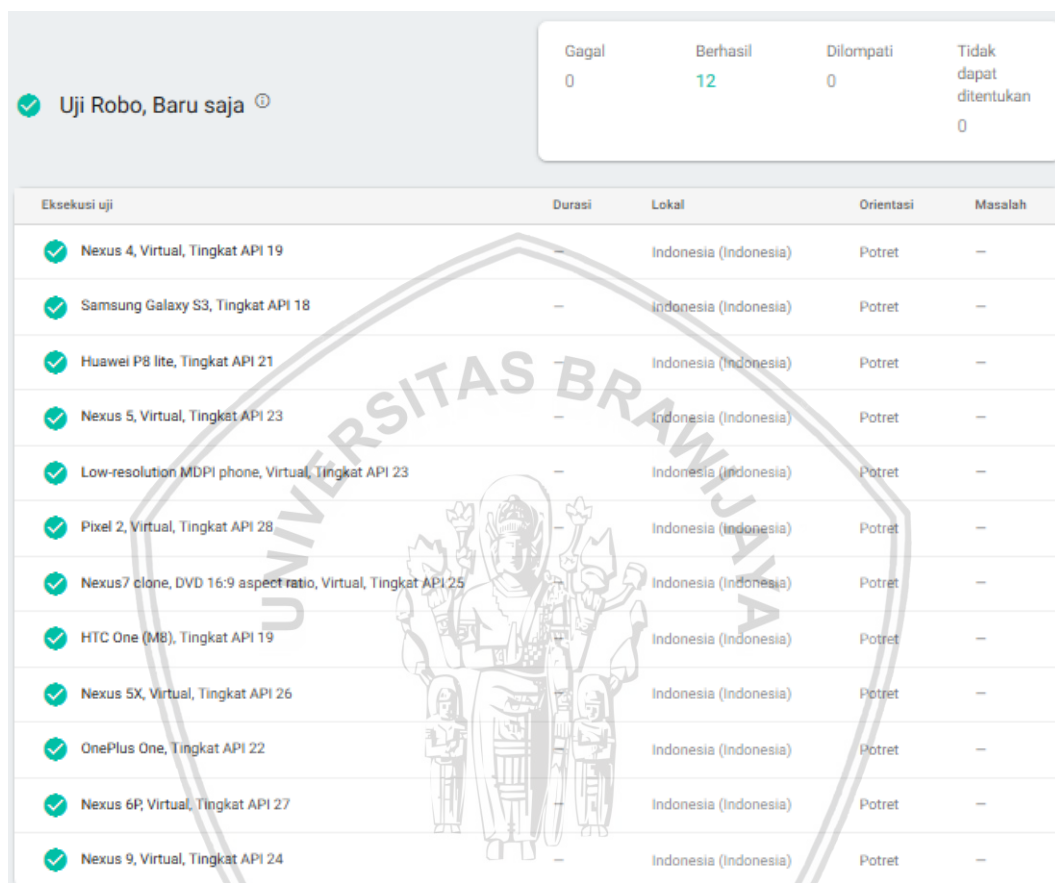
No.	Pengujian	Harapan	Hasil
19	Berhasil menghapus produk	Sistem berhasil menghapus produk	<i>Valid</i>
20	Batal menghapus produk	Sistem tidak menghapus produk	<i>Valid</i>
21	menampilkan total hasil pemasukan dan total pemesanan pada hari ini	Sistem menampilkan total hasil pemasukan dan total pemesanan pada hari ini	<i>Valid</i>
22	menampilkan total hasil pemasukan dan total pemesanan pada tanggal tertentu	Sistem menampilkan total hasil pemasukan dan total pemesanan pada tanggal tertentu	<i>Valid</i>
23	menampilkan total hasil pemasukan dan total pemesanan pada rentang tanggal tertentu	Sistem menampilkan total hasil pemasukan dan total pemesanan pada rentang tanggal tertentu	<i>Valid</i>
24	Terdapat produk yang dijual	Sistem menampilkan list produk yang dijual penjual	<i>Valid</i>
25	Tidak ada produk penjualan	Sistem tidak menampilkan list produk yang dijual penjual	<i>Valid</i>
26	Berhasil menambah pesanan	Sistem berhasil menambah pesanan, mengurangi stok produk dan menampilkan total harga beserta tombol "Pesan Sekarang"	<i>Valid</i>
27	Menambah pesanan pada produk yang sama	Sistem berhasil ngubah jumlah pesanan dan total harga, serta mengurangi stok produk	<i>Valid</i>

No.	Pengujian	Harapan	Hasil
28	Mengurangi produk pesanan	Sistem berhasil ngubah jumlah pesanan dan total harga, serta menambah stok produk	<i>Valid</i>
29	Menghapus produk pesanan	Sistem berhasil menghapus produk pesanan, serta menambah stok produk	<i>Valid</i>
30	Berhasil membayar pesanan	Sistem berhasil mengubah status pesanan menjadi telah dibayar	<i>Valid</i>
31	Gagal membayar pesanan	Sistem memberi peringatan jika saldo tidak cukup	<i>Valid</i>
32	Berhasil mengisi saldo	Sistem menampilkan detail transaksi dan berhasil menambahkan saldo sesuai nominal yang diinput	<i>Valid</i>
33	Gagal mengisi saldo	Sistem memberi peringatan harus mengisi semua inputan	<i>Valid</i>
34	Menampilkan data profil pengguna	Sistem berhasil mendapatkan dan menampilkan informasi profil pengguna dari season	<i>Valid</i>
35	Login berhasil menggunakan username penjual	Sistem masuk ke tampilan penjual	<i>Valid</i>
36	Login berhasil menggunakan username pembeli	Sistem masuk ke tampilan pembeli	<i>Valid</i>

No.	Pengujian	Harapan	Hasil
37	Username dan password kosong	Sistem memberi peringatan inputan harus diisi	<i>Valid</i>
38	Username dan password salah	Sistem memberi peringatan username dan password salah	<i>Valid</i>
39	Registrasi berhasil	Sistem berhasil membuat user pembeli baru	<i>Valid</i>
40	Gagal karena username telah ada	Sistem menampilkan dialog peringatan username yang diinputkan sudah terdaftar sebelumnya	<i>Valid</i>
41	Gagal karena email salah	Sistem memberi peringatan email salah	<i>Valid</i>
42	Gagal karena inputan kosong	Sistem memberi peringatan inputan harus diisi	<i>Valid</i>
43	Mematikan tombol <i>switch</i>	Status toko akan berubah menjadi tutup	<i>Valid</i>
44	Menghidupkan tombol <i>switch</i>	Status toko akan berubah menjadi buka	<i>Valid</i>
45	Membatalkan pesanan	Status pesanan berubah menjadi dibatalkan	<i>Valid</i>
46	Tidak jadi membatalkan pesanan	Status pesanan tidak berubah.	<i>Valid</i>
47	Berhasil mengambil pesanan	Status pesanan berubah menjadi diambil.	<i>Valid</i>

## 7.2 Pengujian *Compatibility*

Pengujian *compatibility* menggunakan Test Lab yang telah disediakan oleh firebase. Firebase Test Lab adalah infrastruktur pengujian aplikasi berbasis cloud. Dengan satu operasi, Anda bisa menguji aplikasi Android atau iOS di berbagai perangkat dan konfigurasi perangkat, dan melihat hasilnya, termasuk log, video, dan screenshot, di Firebase console (Firebase, 2018).



Uji Robo, Baru saja		Gagal	Berhasil	Dilompati	Tidak dapat ditentukan
		0	12	0	0

Eksekusi uji	Durasi	Lokal	Orientasi	Masalah
✓ Nexus 4, Virtual, Tingkat API 19	—	Indonesia (Indonesia)	Potret	—
✓ Samsung Galaxy S3, Tingkat API 18	—	Indonesia (Indonesia)	Potret	—
✓ Huawei P8 lite, Tingkat API 21	—	Indonesia (Indonesia)	Potret	—
✓ Nexus 5, Virtual, Tingkat API 23	—	Indonesia (Indonesia)	Potret	—
✓ Low-resolution MDPI phone, Virtual, Tingkat API 23	—	Indonesia (Indonesia)	Potret	—
✓ Pixel 2, Virtual, Tingkat API 28	—	Indonesia (Indonesia)	Potret	—
✓ Nexus7 clone, DVD 16:9 aspect ratio, Virtual, Tingkat API 25	—	Indonesia (Indonesia)	Potret	—
✓ HTC One (M8), Tingkat API 19	—	Indonesia (Indonesia)	Potret	—
✓ Nexus 5X, Virtual, Tingkat API 26	—	Indonesia (Indonesia)	Potret	—
✓ OnePlus One, Tingkat API 22	—	Indonesia (Indonesia)	Potret	—
✓ Nexus 6P, Virtual, Tingkat API 27	—	Indonesia (Indonesia)	Potret	—
✓ Nexus 9, Virtual, Tingkat API 24	—	Indonesia (Indonesia)	Potret	—

**Gambar 7.1** Hasil pengujian *Compatibility* menggunakan Test Lab Firebase

Hasil dari pengujian pada Gambar 7.1 menunjukkan bahwa dari 12 perangkat yang berbeda telah berhasil dijalankan tanpa ada kegagalan dalam proses *running*. Dapat dilihat juga ada satu perangkat dengan resolusi rendah juga dapat dijalankan tanpa ada masalah.

## 7.3 User Acceptance Testing

Pelaksanaan *user acceptance testing* (UAT) dilakukan dengan memberikan penjelasan mengenai sistem yang akan diuji untuk memberikan gambaran awal bagaimana sistem tersebut. Langkah berikutnya adalah meminta responden melakukan serangkaian task yang telah diberikan. Responden tidak diperkenankan untuk melakukan task diluar dari skenario yang diberikan. Responden memberikan penilaian terhadap pernyataan yang telah disiapkan setelah melakukan rangkaian task yang diberikan. Berikut merupakan skenario task UAT

yang dapat dilihat pada Tabel 7.3 untuk UAT Pembeli dan Tabel 7.4 untuk UAT Pihak Kantin.

**Tabel 7.3 Skenario task *User Acceptance Testing* pada Pembeli**

No.	Tugas yang dikerjakan
1	Melakukan login untuk masuk kedalam sistem
2	Memilih menu profil
3	Menekan tombol “+” untuk tambah saldo
4	Mengisi form inputan tambah saldo
5	Menekan tombol “tambah”
6	Pergi ke Menu Penjualan
7	Menekan tombol “add” untuk melakukan pemesanan
8	Menekan tombol “Pesan Sekarang” untuk pergi ke Menu Pembayaran
9	Menekan tombol “Bayar Sekarang” untuk melakukan pembayaran
10	Pergi ke menu Riwayat Pesanan untuk melihat status pesanan

**Tabel 7.4 Skenario task *User Acceptance Testing* pada Pihak Kantin**

No.	Tugas yang dikerjakan
1	Melakukan login untuk masuk kedalam sistem
2	Pergi ke Kelola Produk
3	Tekan tombol “+” untuk tambah produk
4	Mengisi form inputan tambah produk
5	Tekan tombol “Tambahkan”
6	Pergi ke menu Pesanan
7	Tekan tombol “selesai” untuk mengubah status pesanan menjadi selesai
8	Tekan tombol “diambil” untuk mengubah status pesanan menjadi telah diambil

Setelah melakukan serangkaian task diatas responden diminta untuk mengisi jawaban sesuai dengan pernyataan yang telah disiapkan dan yang sesuai dengan pengalaman pengguna dalam mencoba beberapa task tersebut. Pernyataan yang diberikan berdasar pada 4 kriteria yang ada pada UAT untuk Pembeli dan pihak Kantin. Semua jawaban pernyataan tersebut mengacu pada skala likert yang terdapat 5 jawaban. Berikut merupakan pernyataan UAT yang dapat dilihat pada Tabel 7.5 untuk Pembeli dan Tabel 7.6 untuk pihak Kantin.



Tabel 7.5 Pernyataan UAT untuk Pembeli

No.	Kriteria	Penjelasan	Pertanyaan
1	<i>Functional Correctness and Completeness</i>	Menggambarkan performa dari sistem yang ada dengan mendefinisikan kedalam parameter yang dapat diukur.	Sistem informasi pemesanan dan Penjualan pada kantin Filkom dapat membantu dalam melakukan pemesanan tanpa harus pergi ke kantin filkom.
			Penjualan produk yang ditampilkan dapat membantu dalam mendapatkan informasi produk yang dijual dan harga produk tanpa harus pergi ke kantin filkom.
			Fitur saldo dapat membantu dalam menangani pembayaran secara langsung tanpa harus pergi ke kantin filkom.
2	<i>Usability</i>	Untuk mengetahui apakah sistem mudah untuk digunakan dan dipelajari.	Interaksi pada sistem informasi pemesanan dan Penjualan pada kantin Filkom mudah digunakan.
			Informasi yang ditampilkan jelas dan mudah dipahami.
3	<i>Performance</i>	Menggambarkan bagaimana sistem yang telah dibuat sudah sesuai dengan yang diharapkan oleh pengguna.	Data pesanan yang dikirim dapat diterima penjual secara <i>realtime</i> / dengan rentang waktu yang singkat.
4	<i>Accuracy</i>	Akurasi digunakan untuk mengetahui sejauh mana tingkat ketepatan nilai yang dihitung dengan nilai yang diharapkan.	Kalkulasi total pembayaran dan kembalian telah sesuai dengan perhitungan sebenarnya.
			Kalkulasi penambahan saldo E-Dompet telah sesuai dengan perhitungan sebenarnya.

Tabel 7.6 Pernyataan UAT untuk Pihak Kantin

No.	Kriteria	Penjelasan	Pernyataan
1	<i>Functional Correctness and Completeness</i>	Menggambarkan performa dari sistem yang ada dengan mendefinisikan kedalam parameter yang dapat diukur.	Sistem informasi pemesanan dan Penjualan pada kantin Filkom dapat membantu dalam menangani pesanan Pembeli.
			Fitur kelola produk penjualan dapat membantu dalam melakukan penjualan kepada pembeli.
			Fitur rekap hasil penjualan dapat menangani perekapan hasil pemasukan dan pesanan.
2	<i>Usability</i>	Untuk mengetahui apakah sistem mudah untuk digunakan dan dipelajari.	Interaksi pada sistem informasi pemesanan dan Penjualan pada kantin Filkom mudah digunakan.
			Informasi yang ditampilkan jelas dan mudah dipahami.
3	<i>Performance</i>	Menggambarkan bagaimana sistem yang telah dibuat sudah sesuai dengan yang diharapkan oleh pengguna.	Data pesanan yang diterima secara <i>realtime</i> / dengan rentang waktu yang singkat.
4	<i>Accuracy</i>	Akurasi digunakan untuk mengetahui sejauh mana tingkat ketepatan nilai yang dihitung dengan nilai yang diharapkan.	Kalkulasi total pemasukan dan total pesanan telah sesuai dengan perhitungan sebenarnya.

### 7.3.1 Hasil *User Acceptance Testing* pada Pembeli

Setelah dilakukan proses pengujian UAT oleh pembeli sebanyak 30 orang yang meliputi mahasiswa Filkom UB dan pihak kantin yaitu pengelola kantin. Langkah selanjutnya adalah melakukan analisis terhadap hasil *user accepted testing*. Berikut merupakan hasil jawaban dari pernyataan UAT yang telah diisi oleh responden.

**Tabel 7.7 Hasil User Acceptance Testing pada Pembeli**

No.	Kriteria	Jumlah pernyataan	Jumlah Jawaban				
			STS	TS	N	S	SS
1	<i>Functional Correctness and Completeness</i>	90	0	1	7	42	40
3	<i>Performance</i>	30	0	0	4	15	11
2	<i>Accuracy</i>	60	0	0	5	22	33
4	<i>Usability</i>	60	0	2	8	33	17
Total		240	0	3	24	112	101

Pada Tabel 7.7 terdapat total 240 pernyataan yang telah dijawab oleh responden pembeli. Hasil keseluruhan jawaban yang didapat adalah 3 pernyataan dijawab Tidak Setuju, 24 pernyataan dijawab Netral, 112 pernyataan dijawab Setuju dan 101 pernyataan dijawab Sangat Setuju. Setelah mengumpulkan data UAT selanjutnya melakukan pengolahan data dengan cara memberi bobot nilai pada skala likert. penilaian data UAT tersebut dapat dilihat pada Tabel 7.8.

**Tabel 7.8 Penilaian Bobot User Acceptance Testing pada Pembeli menurut skala Likert**

No.	Kriteria	Nilai Maksimal	Nilai Bobot				
			STS	TS	N	S	SS
	<i>Functional Correctness and Completeness</i>	90 x 5	0 x 1	1 x 2	7 x 3	42 x 4	40 x 5
	<i>Performance</i>	30 x 5	0 x 1	0 x 2	4 x 3	15 x 4	11 x 5
	<i>Accuracy</i>	60 x 5	0 x 1	0 x 2	5 x 3	22 x 4	33 x 5
	<i>Usability</i>	60 x 5	0 x 1	2 x 2	8 x 3	33 x 4	17 x 5
Total Nilai		1200	0	6	72	448	505

Dengan didapatkannya hasil penilaian bobot UAT, dapat dilanjutkan pada tahap berikutnya yaitu pengolahan nilai bobot UAT menjadi suatu presentase penerimaan pengguna. Hasil pengolahan tersebut dapat dilihat pada Tabel 7.9.

**Tabel 7.9 Hasil pengolahan data User Acceptance Testing pada Pembeli**

No.	Kriteria	Persentase Penerimaan
1	<i>Functional Correctness and Completeness</i>	86,89%
2	<i>Performance</i>	84,67%

No.	Kriteria	Persentase Penerimaan
3	<i>Accuracy</i>	89,33%
4	<i>Usability</i>	81,67%
Total Persentase Penerimaan		85,92%

Dapat dilihat total hasil presentase penerimaan untuk pembeli didapatkan 85,92% responden menerima sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB. Dari hasil tersebut juga dapat dilihat presentase penerimaan pada kriteria *Functional Correctness and Completeness* sebesar 86,89%, kriteria *Performance* sebesar 84,67%, kriteria *Accuracy* sebesar 89,33%, dan pada kriteria *Usability* sebesar 81,67%.

### 7.3.2 Hasil *User Acceptance Testing* pada Pihak Kantin

Setelah selesai melakukan analisis terhadap hasil *user accepted testing* pada pembeli selanjutnya yaitu melakukan analisis hasil *user accepted testing* pada pihak kantin Filkom UB. Berikut merupakan hasil jawaban dari pernyataan UAT yang telah diisi oleh responden.

**Tabel 7.10 Hasil *User Acceptance Testing* pada Pihak Kantin**

No.	Kriteria	Jumlah pernyataan	Jumlah Jawaban				
			STS	TS	N	S	SS
1	<i>Functional Correctness and Completeness</i>	3	0	0	0	1	2
3	<i>Performance</i>	1	0	0	0	1	0
2	<i>Accuracy</i>	1	0	0	0	1	0
4	<i>Usability</i>	2	0	0	1	1	0
Total		7	0	0	1	4	2

Pada Tabel 7.10 terdapat total 7 pernyataan yang telah dijawab oleh responden pihak kantin. Hasil keseluruhan jawaban yang didapat adalah 1 pernyataan dijawab Netral, 4 pernyataan dijawab Setuju dan 2 pernyataan dijawab Sangat Setuju. Setelah mengumpulkan data UAT selanjutnya melakukan pengolahan data dengan cara memberi bobot nilai pada skala likert. penilaian data UAT tersebut dapat dilihat pada Tabel 7.11.

**Tabel 7.11 Penilaian Bobot *User Acceptance Testing* pada Pihak Kantin menurut skala Likert**

No.	Kriteria	Nilai Maksimal	Nilai Bobot				
			STS	TS	N	S	SS
	<i>Functional Correctness and Completeness</i>	3 x 5	0 x 1	0 x 2	0 x 3	1 x 4	2 x 5
	<i>Performance</i>	1 x 5	0 x 1	0 x 2	0 x 3	1 x 4	0 x 5
	<i>Accuracy</i>	1 x 5	0 x 1	0 x 2	0 x 3	1 x 4	0 x 5
	<i>Usability</i>	2 x 5	0 x 1	0 x 2	1 x 3	1 x 4	0 x 5
Total Nilai		35	0	0	3	16	10

Dengan didapatkannya hasil penilaian bobot UAT, dapat dilanjutkan pada tahap berikutnya yaitu pengolahan nilai bobot UAT menjadi suatu presentase penerimaan pengguna. Hasil pengolahan tersebut dapat dilihat pada Tabel 7.9.

**Tabel 7.12 Hasil pengolahan data *User Acceptance Testing* pada Pihak Kantin**

No.	Kriteria	Persentase Penerimaan
1	<i>Functional Correctness and Completeness</i>	93,33%
2	<i>Performance</i>	80%
3	<i>Accuracy</i>	80%
4	<i>Usability</i>	70%
Total Persentase Penerimaan		82,86%

Dapat dilihat total hasil presentase penerimaan untuk pihak kantin didapatkan 82,86% responden menerima sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB. Dari hasil tersebut juga dapat dilihat presentase penerimaan pada kriteria *Functional Correctness and Completeness* sebesar 93,33%, kriteria *Performance* sebesar 80%, kriteria *Accuracy* sebesar 80%, dan pada kriteria *Usability* sebesar 70%.

Sehingga hasil dari *user accepted testing* yang melibatkan 30 responden pembeli dengan total 240 jawaban dan 1 responden pihak kantin dengan total 7 jawaban. Dapat diambil kesimpulan bahwa sebesar 85,92% pada pembeli dan 82,86% pada pihak kantin, menyatakan "SETUJU" bahwa sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB dapat membantu pemesanan dan penjualan di kantin. Dari hasil pengujian ini juga dapat dijadikan acuan bahwa operasional CRM yang diterapkan dapat berjalan dan mendapatkan respon yang positif.

## 7.4 Pengujian Waktu

Dari hasil analisis perbaikan proses bisnis *to-be* atau proses bisnis usulan memberikan solusi dan peningkatan dari proses bisnis *as-is*. Pengujian proses bisnis *to-be* menggunakan perbandingan waktu dari proses bisnis *as-is* dengan proses bisnis *to-be*. Berikut merupakan analisis waktu dari proses bisnis *to-be*:

**Tabel 7.13 Pengujian Waktu pada Proses Bisnis Pemesanan dan Penjualan pada Kantin Filkom (to-be)**

No.	Aktivitas	Resp. 1	Resp. 2	Resp. 3
		Waktu(menit)	Waktu(menit)	Waktu(menit)
1	Melihat Daftar Menu Makanan dan Minuman sampai Melakukan Pembayaran	00:30.6	00:30.7	00:30.1
2	Menunggu Pesanan Selesai ( <i>intermediate event</i> menerima notifikasi pesanan telah selesai)	06:06.0	06:19.0	06:00.0
3	Menunjukkan Bukti Pemesanan ke Penjual sampai Mengambil Pesanan	01:24.0	01:33.0	01:35.0

Tabel 7.13 menunjukkan bahwa hasil pengujian waktu dari 3 responden terdapat total waktu kurang lebih 8 menit 30 detik. Jika dibandingkan dengan waktu pada proses bisnis *as-is* yang sebesar 13 menit maka dapat disimpulkan bahwa pada proses bisnis *to-be* lebih cepat 4 menit 30 detik dari proses bisnis *as-is*.



## BAB 8 PENUTUP

### 8.1 Kesimpulan

Terdapat beberapa kesimpulan yang dapat diambil dari hasil penelitian yang dilakukan sejauh ini, yaitu:

1. Pada fase *inception* dilakukan analisis proses bisnis pemesanan pada kantin filkom saat ini, lalu dari analisis tersebut ditemukan 4 permasalahan yaitu penjual yang kebingungan dalam melayani pembeli jika ramai, proses pemesanan yang manual, penjual akan kebingungan untuk mencari pembeli yang menunggu digazebo, pembeli yang harus mengantri didepan stan. Dengan menggunakan konsep Operasional CRM khususnya pada *Sales Force Automation* dan *Service Automation* ditemukan solusi dari permasalahan tersebut. Dengan solusi tersebut dapat dibuat untuk perbaikan proses bisnis tersebut agar dapat diterapkan dan dibuatkan sistem yang sesuai dengan solusi tersebut. Hasil yang didapat ialah sebuah rekomendasi proses bisnis pemesanan pada kantin filkom (*To-be*) dan tambahan proses bisnis dalam melakukan kelola produk penjualan.
2. Fase *Elaboration* merupakan fase selanjutnya yang melakukan tahapan analisis kebutuhan dan juga mendesain perancangan sistem. Hasil yang didapat dari analisis kebutuhan ialah peneliti mendapatkan berbagai kebutuhan dari hasil perbaikan proses bisnis yang didapat dalam fase *inception* dan melakukan desain *use-case diagram* untuk menggambarkan fungsional-fungsional sistem yang akan dibuat. Fase elaboration pada iterasi selanjutnya terdapat penambahan 2 fungsional sistem, jadi total fungsional yang didapat adalah 14, dan tahap selanjutnya yaitu perancangan sistem. Pada perancangan sistem dihasilkan Activity Diagram yang merupakan alur interaksi pengguna terhadap sistem, Sequence Diagram yang merupakan alur interaksi antar obyek sistem yang akan dibuat, dan dari hasil pemodelan sequence diagram tersebut didapatkan 24 kelas. Selanjutnya ialah melakukan perancangan *physical data model*, antarmuka pengguna dan melakukan perencanaan implementasi.
3. Pada fase *Construction* terdapat perubahan sedikit pada perancangan *physical data model* karena terdapat ketidak efisienan rancangan tersebut, dan memulai tahap implementasi. Hasil dari perubahan rancangan *physical data model* yaitu pemecahan data model user menjadi penjual dan pembeli. Hasil dari tahap implementasi yaitu hasil implementasi Basis Data yang menggunakan Firebase Realtime Database, hasil implementasi kelas yang terdiri dari 14 kelas yang berperan sebagai *view*, 5 kelas *controller* dan 5 kelas *model* dan 7 kelas untuk *physical data model*, dan hasil dari implementasi antarmuka pengguna. Iterasi selanjutnya terdapat penambahan 2 fungsional yang mengakibatkan penambahan pada rancangan-rancangan sebelumnya juga dan melakukan pengujian sistem

menggunakan pengujian *Black Box* dengan menggunakan 47 *test case* dan pengujian komparabilitas menggunakan *tools Test Lab*. Hasil pengujian pada *black box* dinyatakan berhasil semua dan pada pengujian komparabilitas dilakukan pengujian pada 12 perangkat yang berbeda dengan menggunakan *Test Lab* dan dari pengujian tersebut dinyatakan berhasil semua.

4. Pada fase *Transition* untuk menerapkan sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB dilakukan *User Acceptance Testing* (UAT) agar dapat dibuktikan bahwa sistem yang dikembangkan dapat diterima oleh pengguna atau tidak serta Pengujian Waktu untuk memastikan proses bisnis *to-be* dapat mempercepat alur pemesanan atau tidak. Hasil dari *user accepted testing* (UAT) yang melibatkan 30 responden pembeli dengan total 240 jawaban dan 1 responden pihak kantin dengan total 7 jawaban. Hasilnya menunjukkan bahwa tingkat penerimaan dari pembeli adalah 85,92% dan 82,86% pada sisi kantin Filkom. Pada Pengujian Waktu dengan perbandingan proses bisnis *as-is* dengan proses bisnis *to-be*, dapat disimpulkan bahwa pada proses bisnis *to-be* lebih cepat 4 menit 30 detik dari proses bisnis *as-is*. Dari hasil pengujian tersebut, dengan menerapkan sistem informasi ini dapat mempermudah dan mempercepat alur pemesanan dan penjualan pada Kantin Filkom.

## 8.2 Saran

Saran yang dapat diberikan setelah melakukan penelitian pengembangan sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB yaitu:

1. Pada pengembangan lebih lanjut dapat diterapkan sistem informasi untuk pengelola kantin agar pengelola kantin dapat memantau perkembangan kantin secara langsung tanpa harus meminta laporan pada penjual dan kasir, dan dapat melakukan penerimaan stan jika membutuhkan stan baru dikantin.
2. Dalam implementasi lebih lanjut dapat dikembangkan lagi sistem untuk menangani pembayaran pada kantin yang berupa *topup* saldo dengan berbagai metode, serta peningkatan keamanan sistem informasi dan *User Experience* pada sistem informasi *sales force automation* dan *service automation* pemesanan dan penjualan pada kantin Filkom UB.

## DAFTAR PUSTAKA

- Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modeling Language User Guide*. Boston: Addison Wesley.
- Buttle, F., 2009. *Customer Relationship Management: Concepts and Technology*. 2nd ed. Oxford: Butterworth-Heinemann.
- Carissa, A. O., Fauzi, A. & Kumadji, S., 2014. Analisa Perancangan Customer Relationship Management (CRM) Poin Pelanggan Pada Biro Komunikasi Pemasaran PT. Semen Indonesia (Persero) Tbk. *Jurnal Administrasi Bisnis*, Volume 15.
- Firestore, 2017. *Firestore Realtime Database*. [Online] Available at: <https://firebase.google.com/docs/database/> [Diakses 19 Januari 2018].
- IBM, D. T., 1998. *Rational Unified Process, Best Practices for Software*. Cupertino: Rational Software.
- Imaduddin, A. & Permana, S., 2017. *Menjadi Android Developer Expert*. Bandung: PT. Presentologics.
- Ishami, F., Rokhmawati, R. I. & Saputra, M. C., 2018. Pengembangan Sistem Customer Relationship Management (CRM) Self-Service Reservation Pada Everyday Smart Hotel Malang Menggunakan Usability Testing. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 632-640.
- Kendall, K. E. & Kendall, J. E., 2011. *System Analysis And Design*. 8th ed. Upper Saddle River: Prentice Hall.
- Naik, K. & Tripathy, P., 2008. *Software Testing And Quality Assurance Theory And Practice*. Hoboken: John Wiley & Sons.
- Nasir, S. Z. & Ahmed, S. M., 2013. *Sales Force Automation Using Mobile Technology*. Gueongju, IEEE.
- O'Brien, J. A. & Marakas, G. M., 2013. *INTRODUCTION TO INFORMATION SYSTEM*. 16 ed. New York: McGraw-Hill.
- Payne, A., 2005. *Handbook of CRM: Achieving Excellence in Customer Management*. Oxford: Butterworth-Heinemann.
- Pressman, R. S., 2010. *Software Engineering - A practitioner's approach*. 7th ed. Boston: McGraw-Hill.
- Silver, B., 2012. *BPMN 2.0 Handbook*. 2nd ed. Florida: FutStrat.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. Boston: Addison-Wesley.
- Sugiyono, 2017. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Usmadi, A. J. M. & Handoko, Y., 2016. Perancangan Customer Relationship Management untuk Membangun Loyalitas Pelanggan pada Divisi Pemasaran PT Bio Farma (Persero). *Jurnal Tata Kelola dan Kerangka Kerja Teknologi Informasi*, Volume 2.

- Weske, M., 2007. *Business Process Management Concepts, Languages, Architectures*. New York: Springer-Verlag Berlin Heidelberg.
- Whitten, J. L. & Bentley, L. D., 2007. *System Analysis and Design Methods*. 7th ed. New York: McGraw-Hill.

